January 1987 UILU-ENG-87-2204

COORDINATED SCIENCE LABORATORY
College of Engineering

EXTRACTING SURFACES FROM STEREO IMAGES: AN INTEGRATED APPROACH

William Hoff
Narendra Ahuja

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-87-2204	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois	6b. OFFICE SYMBOL (If applicable) N/A
7a. NAME OF MONITORING ORGANIZATION National Science Foundation	
6c. ADDRESS (City, State and ZIP Code) 1101 W. Springfield Avenue Urbana, Illinois 61801	7b. ADDRESS (City, State and ZIP Code) 1800 G. Street, N.W. Washington, D.C. 20550
8a. NAME OF FUNDING/SPONSORING ORGANIZATION National Science Foundation	8b. OFFICE SYMBOL (If applicable) N/A
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NSF ECS-83-52408	
8c. ADDRESS (City, State and ZIP Code) 1800 G. Street, N.W. Washington, D.C. 20550	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
N/A	N/A	N/A	N/A

11. TITLE (Include Security Classification) "Extracting Surfaces from Stereo Images: An Integrated Approach"

12. PERSONAL AUTHOR(S) William Hoff and Narendra Ahuja

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM _____ TO _____	January 1987	118

16. SUPPLEMENTARY NOTATION
N/A

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	stereo vision, surface interpolation, feature matching, contour detection, three-dimensional depth

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Stereo vision provides the capability of determining three-dimensional distance of objects from a stereo pair of images. The usual approach is to first identify corresponding features between the two images, then interpolate to obtain a complete distance or depth map. Traditionally, finding the corresponding features has been considered to be the most difficult problem. Also, occluding and ridge contours (depth and orientation discontinuities) have not been explicitly detected and this has made surface interpolation difficult. The approach described in this paper is novel in that it integrates the processes of feature matching, contour detection, and surface interpolation. Integration is necessary to ensure that the detected surface is smooth. The surface interpolation process takes into account the detected occluding and ridge contours in the scene; interpolation is performed within regions enclosed by these contours. Planar and quadratic patches are used as local models of the surface. Occluded regions in the image are identified and are not used for matching and interpolation.

The approach described is fairly domain-independent since it uses no constraint other than the assumption of piecewise smoothness. A coarse-to-fine algorithm is presented that requires no human intervention other than an initial rough estimate of depth. The surface estimate obtained at any given level of resolution is used to predict the expected locations of the matches at the next finer level. As the final result, a multiresolution hierarchy of surface maps is generated, one at each level of resolution. Experimental results are given for a variety of stereo images.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
		NONE

Extracting Surfaces from Stereo Images: An Integrated Approach

William Hoff
Narendra Ahuja

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1101 W. Springfield Ave.
Urbana, Illinois 61801

ABSTRACT

Stereo vision provides the capability of determining three-dimensional distance of objects from a stereo pair of images. The usual approach is to first identify corresponding features between the two images, then interpolate to obtain a complete distance or depth map. Traditionally, finding the corresponding features has been considered to be the most difficult problem. Also, occluding and ridge contours (depth and orientation discontinuities) have not been explicitly detected and this has made surface interpolation difficult. The approach described in this paper is novel in that it integrates the processes of feature matching, contour detection, and surface interpolation. Integration is necessary to ensure that the detected surface is smooth. The surface interpolation process takes into account the detected occluding and ridge contours in the scene; interpolation is performed within regions enclosed by these contours. Planar and quadratic patches are used as local models of the surface. Occluded regions in the image are identified and are not used for matching and interpolation.

The approach described is fairly domain-independent since it uses no constraint other than the assumption of piecewise smoothness. A coarse-to-fine algorithm is presented that requires no human intervention other than an initial rough estimate of depth. The surface estimate obtained at any given level of resolution is used to predict the expected locations of the matches at the next finer level. As the final result, a multiresolution hierarchy of surface maps is generated, one at each level of resolution. Experimental results are given for a variety of stereo images.

The support of the National Science Foundation under grant ECS 8352408 and Rockwell International is gratefully acknowledged.

1. INTRODUCTION

The goal of stereo vision is the recovery of three dimensional depth information from images taken from different viewpoints. In this paper we compute the distance between the viewer and each point of the scene visible from two viewpoints, using two images recorded simultaneously from a pair of laterally displaced cameras (Figure 1). It is possible to use more than two images — three are used in [Piet86] and [Ito86] — and the information provided by the additional views is useful. However, the fundamental issues are the same for two camera stereo or for three camera stereo, and we will concentrate on the two camera problem.

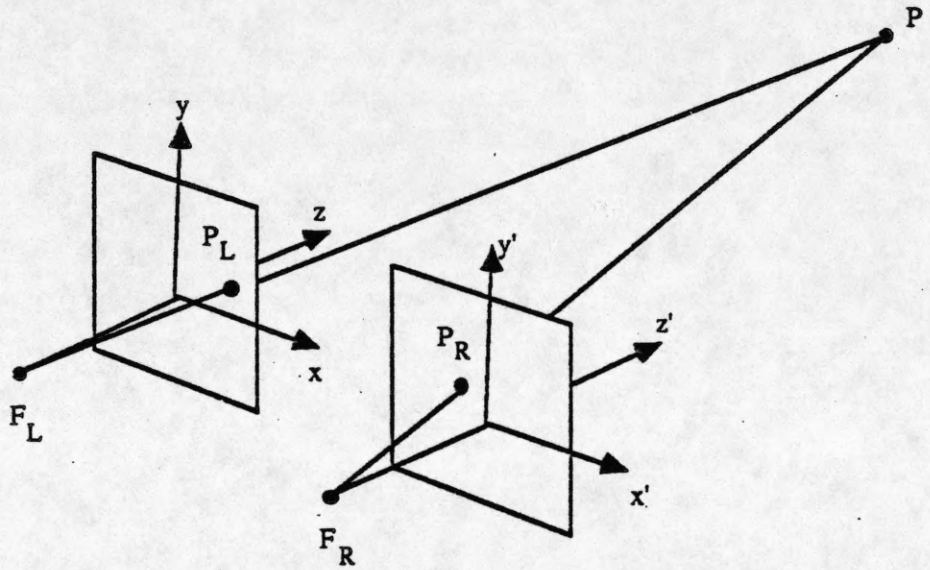

Figure 1. The stereo imaging geometry. Point P projects onto the left and right image planes at points P_L and P_R, respectively.

Although stereo vision has a very general domain of applicability, several requirements should be met for its effective use. First, the images must be sufficently textured so that distinguishing features can be identified, *i.e.* the images must not be of uniform intensity. Second, there must be some displacement between the viewpoints. In general, the accuracy of the reconstructed depth map increases with the displacement between the viewpoints. Third, depth can only be calculated for scene points that are visible to both viewpoints. Thus, the two viewpoints must be sufficiently close so that most of the scene is visible to both viewpoints.

The basis for stereo algorithms is that the distance to a point in the scene can be computed from the relative difference in position of the projection of that point in the two images. This difference is called the *disparity* and is measurable in angular units. The usual paradigm for stereo algorithms includes the following steps:

(1) Features are located in each of the two images independently.

(2) Features from one image are matched with features from the other image. Typically, for every feature in the left image corresponding to a certain point in the scene, a feature must be found in the right image such that it corresponds to the projection of the same scene point.

(3) The disparity between features is used, together with estimates of the parameters of the imaging geometry (*i.e.*, relative separation and orientation of the cameras), to determine the distance to the corresponding point in the scene.

(4) The resulting depth points are often sparse whereas depth must be computed at every point in the scene. Thus, the depth points are interpolated to obtain a surface, or a complete depth map.

The problem is essentially one of simple triangulation if the locations in the two images of each visible scene point are known. However, the correspondences across images must be established first if triangulation is to be used. As demonstrated by Julesz [Jule71], it may be sufficient to consider only syntactic or low level features of local gray level patterns to establish point correspondences across images. Two points having similar gray level contexts, or features, serve as candidates for being projections of a single scene point, if the chosen feature is invariant of change in viewpoint.

Because of their simplicity, similar low level features occur commonly in the image. The search for the match of a point often yields multiple candidates, because there is little information that can be used to characterize low level features uniquely. Thus, there can be many possible matches for a given feature, and it is necessary to choose the correct match from all the false targets. The matching step above incorporates a resolution of this ambiguity. Since the selected matches are crucial in determining the resulting surface map, this step, called the *correspondence problem*, has been considered to be the central and the most difficult part of the stereo problem.

Features are defined over neighborhood intensities and not all image points, in general, may have distinguishable features. This means that the depth values can at best be found for a subset of image points, and a complete surface map must be inferred from these sparsely located depth samples of the three dimensional surfaces. Methods for interpolating a surface from sparse depth constraints are given in [Akim78] and [Terz83].

The following are some problems that all stereo algorithms must deal with:

(1) Ambiguous matches or false targets. The prevalence of matching ambiguities depends on the kind of features matched, and on the characteristics of the scene. Scenes with periodic structures, which are common in man-made environments, give rise to the "wallpaper effect," in which there is a consistent mismatching of features.

(2) Occluding contours, which are places in the scene where the depth changes discontinuously. These create a problem because they may not be intrinsically characteristic of the scene, and vary with viewpoint. Further, it is difficult to resolve ambiguous matches in the vicinity of the contour, if the neighboring disparity values correspond to more than one surface. Another problem with occlusions is that entire regions of feature points are unmatchable, because that part of the object's surface is visible to only one eye or camera.

(3) Mismatched points, which can arise from several causes. A point which is unmatchable due to occlusion may receive an incorrect match. Another cause of mismatches is random noise, which is present in one image but not the other. Finally, photometric effects can cause the light intensity received from the same surface patch to be different at the two cameras. This could cause a feature to be detected in one image but not the other.

The following constraints are often used for stereo matching:

(1) If the cameras are located and oriented so that they are coplanar and there is only a horizontal translational difference between them, then disparity can only occur in the horizontal direction. The search for a corresponding point is then restricted to the corresponding horizontal line in the other image. If the cameras do not have the same orientation, the search can still be restricted to a line in the other image (called the *epipolar* line), but the line is not horizontal. Figure 2 shows the geometry. This constraint, called the epipolar constraint, allows the search space to be reduced from two dimensions to one dimension, with an enormous reduction in computational complexity.

(2) Since the images are taken simultaneously, disparity is caused only by the difference in the two viewpoints, and is determined by the depth of the surface. Therefore, one can take advantage of the fact that most surfaces in the real world are smooth, in the sense that local variations in the surface are small compared with the overall distance from the viewer. It follows that disparity varies smoothly almost everywhere, except at the relatively rare object boundaries. This constraint, called the continuity constraint, is useful in resolving ambiguous matches [Marr82].

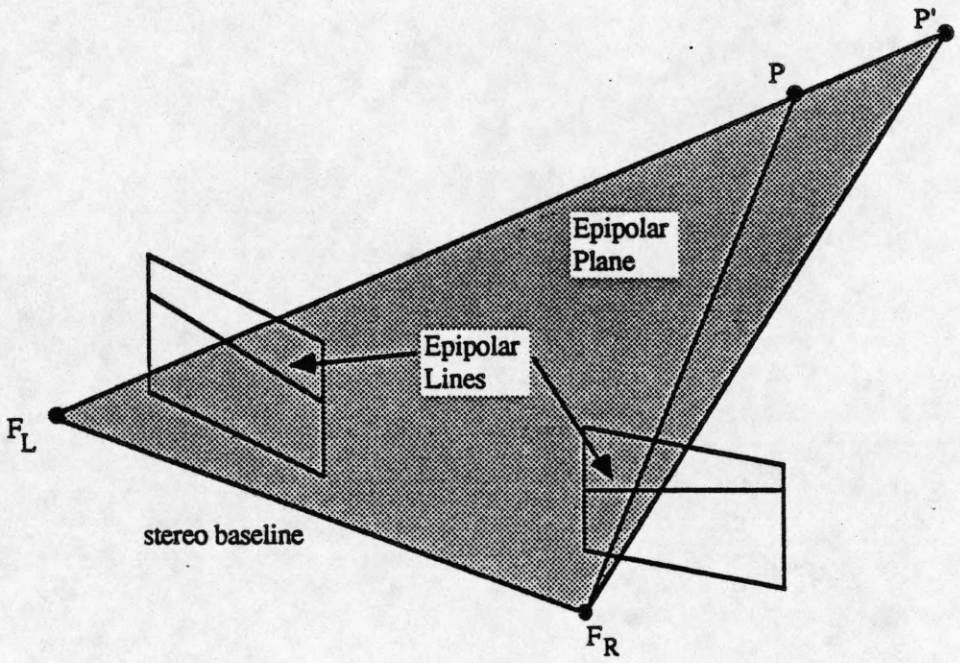

Figure 2. The epipolar constraint. Point P, and the left and right focal points, F_L and F_R determine a unique epipolar plane. The projection of P in the right image lies along the line that is the intersection of this epipolar plane with the right image plane. This line is called the epipolar line. (From [Barn82]).

Both constraints are used by most algorithms, although the interpretation of the continuity constraint is varied. These constraints are used to match features, before any interpolation of the resulting depth values is performed.

This paper argues that the steps of matching and surface interpolation should be merged. An approach to estimating surfaces from stereo is presented that integrates feature matching, ridge and occluding contour detection, and surface interpolation (Section 2). The goal is to perform matching such that the interpolated surface is smooth except across ridge and occluding contours. A coarse-to-fine algorithm is presented that implements the integration to produce a hierarchy of surface maps, corresponding to different levels of resolution (Section 4). Section 5 presents results of the algorithm on several images, and Section 6 gives conclusions.

2. PREVIOUS WORK

In this section, previous work on stereo vision is reviewed. The review covers stereo algorithms that have been implemented and described in the literature. It is not intended to be exhaustive, but is extensive enough to show the state of the art. The review does not cover the psychophysical literature. A critique is then presented, which motivates the present work. Most algorithms have used the epipolar constraint, and some form of the disparity continuity constraint. The primary difference among the algorithms is in the interpretation of the continuity constraint.

2.1 Area-Based Algorithms

Existing algorithms may be classified according to the type of features they detect — edges or patches of area [Bake81]. Edge-based algorithms use intensity edges as features and attempt to match individual edge points [Arno78, Barn80, Grim81, Hend79, Kim86, Ohta85], or linear edge segments which consist of chains of aligned edge points [Ayac85, Medi85]. The area-based approaches use as features the absolute intensity values in the vicinity of a point, say, in a small window centered at the point; the gray level correlation between the windows of two points is often used to evaluate the quality of the match between the points [Genn77, Hann80, Luca82, Mora81, Pant78]. In several of these algorithms ([Genn77], [Hann80], [Mora81]), a few points are first matched with another technique to provide a rough guide for the high-resolution correlation.

Area-based techniques suffer from certain inherent limitations, including the following: First, there is a tradeoff between window size and disparity precision. Second, it is difficult to deal with scenes that contain depth discontinuities, because the windows may cross discontinuities Third, photometric properties of a scene are not invariant to viewing position, as the technique assumes.

2.2 Edge-Based Algorithms

Edge-based approaches avoid some of the disadvantages of the area-based approaches. First, edges are associated with intensity changes rather than absolute values of intensities and are, therefore, a better characteristic of physical changes in the scene. Second, they are intrinsically more localizable, theoretically to sub-pixel precision. Third, edge-based approaches may be faster than area-based approaches because there are fewer features to deal with. Edge features have been more effective and popular in the work on computational stereo, and the remainder of this section concentrates on algorithms that use this approach.

The theory of stereo vision developed by Marr and Poggio [Marr79] was further developed and implemented by Grimson [Grim81, Grim85]. The features used for matching are the zero crossings of the Laplacian of a Gaussian: $\nabla^2 G$ [Marr80]. The search for matching zero crossings is restricted to corresponding rows, because the epipolar lines are assumed to be horizontal. The processing is top-down, so that the result of matching at coarse resolution can be used to guide the matching at the finer levels of resolution. For each zero crossing in the left image at a particular level of resolution, the algorithm searches for possible matches on the corresponding row in the right image, at the same level of resolution, in a window centered at the location predicted by the previous level. A zero crossing is a possible match if it has the same sign as the zero crossing in the left image, and if the contour on which it lies has a similar orientation at that point.

The size of the window is $2w$, where $w = 2\sqrt{2}\sigma$ is the width of the filter used at the current level. From the statistical analysis of the intervals between zero crossings for an image composed of white Gaussian noise, the probability of there being more than one zero crossing of the same sign in the window is less than 50% [Marr79]. Thus, more than 50% of the time, the match will be unambiguous and the remaining cases will be ambiguous, usually with two possible matches. In Grimson's implementation of the algorithm, the window was divided into

three pools: one divergent, with a disparity range of [-w,0); one convergent, with a disparity range of (0,w]; and one with zero disparity. To resolve ambiguities, the algorithm scans a neighborhood about an ambiguous point, recording the sign of the disparities (convergent, divergent, or zero) of the unambiguous matches in the neighborhood. If the ambiguous point has a potential match of the same sign as the dominant type in the neighborhood, then that is chosen as the match. If there are more than two candidates in a single pool, then the point is rejected and no attempt is made to match it. The justification for this method comes from the continuity constraint — the disparities are not expected to vary much over a small neighborhood.

It is possible that the expected disparity of a region is incorrect and that the true disparity is beyond the range of the window being used. In this case, the zero crossings may find matches at random. The probability that a zero crossing from the left image finds a match in the right image is just the probability of a zero crossing of the right sign falling within the window, which is about 0.7. (Actually, the probability is somewhat less than 0.7, because the orientation is considered in addition to the sign for matching.) If the percentage of points in a region that have matches is less than or equal to 0.7, then the region is declared to be out of range, and no disparity values are accepted for that region.

On the examples shown, the Marr-Poggio-Grimson algorithm gives impressive results. However, the following characteristics of the algorithm should be noted:

(1) The neighborhoods used for resolving ambiguities and for detecting out-of-range regions may cross occluding contours. In these situations, the continuity constraint does not hold, because the disparity values in the neighborhood come from two different surfaces. An incorrect match may be chosen, because some of the points in the neighborhood are on a different surface, with unrelated disparities, and yet these are used to determine the majority. When detecting out-of-range regions, the region may cross an occluding contour, causing more than 30% of the points to be out of range of the matching window. Points in the entire window in this case would be rejected, including the matchable points.

(2) The method for resolving ambiguities assumes that the sign of the disparity does not vary over a small neighborhood in the vicinity of the ambiguous point. The method for detecting out-of-range regions assumes that if the region is within range, then the disparities of all points in the region are within the range of the matching window. Both these assumptions may be violated if the surface is sloping or curving, and thus does not have a constant disparity. The cause of this problem is the meaning and use of the continuity constraint. The continuity constraint, as defined by Marr and Poggio, specifies that surfaces are expected to vary smoothly almost everywhere. "Smoothly" means that the surface variation due to roughness cracks or other sharp differences are small compared with the overall distance from the viewer. However, if the surface has some overall slope, then the surface variation may not be insignificant. Effectively, the Marr-Poggio-Grimson algorithm uses the continuity constraint to mean constancy of disparity. The following example poses a problem for such an ambiguity resolution procedure: The surface in the local neighborhood is a simple inclined plane, and the correct match for the ambiguous point in question is close to zero disparity. In this case, most of the surface in the local neighborhood has a disparity unequal to zero, and so the zero disparity match would not be chosen. A problem still exists if the zero disparity pool were eliminated and just the convergent and divergent pools were used. In this case, if the correct match were close to zero disparity, there might happen to be more points in the local neighborhood with the opposite sign, because of non-uniform zero crossing density. This also might cause the incorrect match to be chosen. This problem may occur often, since the matching window is centered at the expected disparity, and therefore the correct match usually has a disparity close to zero.

(3) Transparent surfaces will be difficult to handle, because it is assumed that the depth points come from the same surface. For example, the matches for a transparent surface region would probably be rejected, if less than 70% of

the matches were on one surface. Also, the method for disambiguating points would not work correctly because the points in a local neighborhood would not all come from the same surface.

(4) The ambiguity resolution method will be adversely affected by noise in the disparity values. Recall that the matching windows are centered on the disparity estimate provided by the results of matching at the coarse levels. Therefore, one would expect most matches to be near the center of the window, or equivalently, to have nearly zero displacement relative to the current disparity estimate. If a point is near zero disparity, a small amount of noise could cause it to change its sign. Even assuming a surface of constant disparity, the noise would cause the majority of the neighboring unambiguous points to lie in the wrong pool.

Instead of using windows, Mayhew and Frisby enforce continuity of disparity along edges in the image [Mayh81], which they call the "figural continuity" constraint. The reasoning behind this constraint is that it may be more difficult to ensure that a region of the image corresponds strictly to a single surface, than that an image edge lies along a single surface, since edges reflect changes in the surface topography or the surface photometry. However, it is possible for an image edge to cross an occluding contour. In such cases disparity will not be continuous. Kim and Bovik [Kim86] also use figural continuity constraint to match points along image edges. The disparity of a point along a contour is estimated by interpolating disparities of matched extremal points along the contours.

Grimson implemented a new version of their earlier algorithm [Grim85], incorporating figural continuity as a criterion to eliminate random matches. The algorithm thus avoids the use of matching statistics over a region, and the accompanying problems. The method for resolving ambiguities is similar to the method in the original algorithm, but with two differences. First, a neighborhood in the next coarser level is searched, instead of the current level. Second, instead of choosing the match that has disparity similar to the most points in the neighborhood, it chooses the match that has disparity similar to at least one point in the neighborhood, as long as none of the other alternative matches have disparities similar to any neighboring points.

The figural continuity constraint is superior to the use of regions for matching statistics, in that it avoids the problem of the region overlapping an occluding contour. Use of the constraint detects cases where the matches are out of range and are being matched at random. The following problems occur:

(1) The probabilities of matching the individual points on a contour are not independent.

(2) Contours are assumed to lie on a single surface. However, it is possible for a contour to cross an occluding boundary. In this case, the part of the contour across the boundary may be out of range and mismatched, but because it belongs to a contour that is sufficiently long, it is accepted as correct. The neighborhood about an ambiguous point may cross an occluding boundary, causing an incorrect match to be chosen or the point to be rejected because of support for more than one match.

(3) A sloping surface would cause potential problems, because points in the neighborhood would have disparities different than the disparity of the correct match of the ambiguous point. For example, consider an ambiguous point on a simple inclined plane. In this case, it is likely that all the potential matches for the point will find support in the neighborhood, causing the point not to be assigned a disparity value.

Henderson, Miller, and Grosch [Hend79] implemented an algorithm which was designed for the specific application of aerial photographs of cultural scenes, which typically contain rectilinear structures. The algorithm matches edges on epipolar lines. Those matches which are seen to "persist" over several preceding line analyses are accepted. The algorithm has a number of constraints built into it that limits its applicability. These include: the surfaces in the scene have to be planar, and the edges have to come from straight lines. Also, manual intervention is required initially, and whenever a new edge is encountered.

The algorithm of Barnard and Thompson [Barn80] is unusual in that it does not use the epipolar constraint, thus making it applicable to motion correspondence as well as stereo. First, feature points are selected with the

Moravec operator [Mora81]. For every feature point P in the first image, a set of possible matches is constructed by taking all the feature points in the second image that lie within a maximum distance r of the (x,y) location of P in the first image. Then, a relaxation labelling technique [Rose76] is used to select one of the candidates as the correct match. The constraint for choosing the correct match is that the points in the local neighborhood should have nearly the same disparity as the correct match.

Baker [Bake81] uses a different matching constraint than any of the previous algorithms. The constraint is that the left-to-right ordering of matched edges along corresponding epipolar lines (which are assumed to be horizontal) should be the same in both the left and right images. The edges are matched via a dynamic programming technique taking into account their contrast and slope. Next, the edges are checked for continuity of disparity and removed if they are not consistent. This is effectively the figural continuity constraint. Although the set of matches chosen on each scan line is optimal, the consistency checking procedure removes matches that do not satisfy inter-scan line consistency. Ohta and Kanade [Ohta85] use the same ordering constraint, but they incorporate the interline consistency constraint as a part of the dynamic programming formulation. Thus, their algorithm ensures the consistency of matches across scan lines.

In the algorithm of Medioni and Nevatia [Medi85], line segments are extracted from each image of the stereo pair by the Nevatia-Babu line finder. Candidates for corresponding lines must have similar orientation and contrast and lie (at least partially) in a window which is as wide as the maximum disparity allowed. A two step algorithm matches every line in the left image to one or more in the right, and vice versa (multiple matches may occur because segments may be fragmented, not because of ambiguities). In the first step, the match is found which is most similar to the disparities of possible matches in its neighborhood. These are called "preferred" matches. In the second step, the match is found which is most similar to the disparities of the preferred matches from the first step. The effect of this step is to reevaluate the matches, using the new information provided by the preferred matches.

Ayache and Paverjon [Ayac85] also extract and match linear edge segments. Their algorithm first finds a small set of hypothesized matches, and then propagates the information about the matches to the neighbors of the matched segments. The initial criterion for matching is similarity of length and orientation. The criteria for the propagated matches are again similarity of length and orientation, but also similarity of disparity of the match to that of the match from which it was propagated. If a propagated match gets a different result from two different initial matches, then the result from that initial match is chosen which has the greatest number of matches propagated from it (called the power of prediction).

2.3 Critique: Partial Use of Surface Smoothness

The constraints used by the stereo vision algorithms summarized above to perform ambiguity resolution are derived from some implicit or explicit assumptions about surface shape and its relationship to the image features detected. For example, in the area-based algorithms it is assumed that the surface has a smooth shape. The assumed model of surface shape is used to determine the shapes of the matching windows in the two images over which the intensity correlation is to be performed. In the Marr-Poggio-Grimson approach, the features in a neighborhood are assumed to be projections of markings mainly from the desired surface, and their disparity values are expected to be similar. The matching ambiguity at a point is resolved such that the chosen disparity value is close to the majority in the neighborhood. This is motivated by the property that nearby points on the surface have similar depths. The figural continuity constraint of Baker [Bake81], Ohta and Kanade [Ohta85], and Mayhew and Frisby [Mayh81] is motivated by the assumption that the depth along a marking on a single surface varies smoothly, and hence, the disparity should also vary smoothly. All these constraints are intended to enforce a model of surface smoothness.

However, these constraints only partially capture the desired model, namely, that the scene contains objects with smooth surfaces. Therefore, enforcing such constraints only partially enforces surface smoothness.

There are two problems with the partial constraints described above. First, they make certain assumptions about the relationship of the detected image features to three dimensional features, which may or may not hold. For example, an intensity edge segment may not lie on a single surface, but may cross different surfaces. Thus, disparity will not vary smoothly along the edge segment. This happens when there are no strong intensity edges defining the boundaries of the different surfaces, but rather the intensity edges cross freely from one surface to another. Although the constraint that the disparity should vary smoothly along a surface marking is valid, not all edges in the image may correspond to surface markings and it may not be known which ones do. Likewise, enforcing similarity of disparity over a window will be erroneous if the window contains an occlusion boundary.

The second problem with the use of partial constraints is that even when the above assumptions about the features are met, namely, the edge segments or windows to which the constraints apply come from a single surface, the constraints do not still enforce surface smoothness in a true sense. For example, in the Marr-Poggio-Grimson approach the constraint that nearby disparities in a neighborhood should be similar in value is too weak to enforce smoothness. This is because smoothness is actually determined by not only the values of disparity in an image region but also the spatial distribution of these values. The disparity values may actually span a wide range without violating smoothness as would be the case for a slanted plane in which the disparities of features in the nearest part are larger than those of features located in the distant part. If it is required that the disparities in a model have similar values, *i.e.*, that the histogram of these local disparities be uniform, then disparity selection is biased in favor of a frontoparallel surface.. The constraint that disparity vary smoothly along a single edge segment along a surface, correctly but weakly enforces the constraint of three dimensional surface smoothness. Three dimensional surface smoothness actually implies a stronger, two-dimensional smoothness in the image; by enforcing smoothness along edges in the image only subsets of local disparities (along curves) are constrained. (The algorithm of Ohta and Kanade [Ohta85] enforces partial two-dimensional smoothness in the image, in that the ordering of edge contours from top to bottom is preserved.)

3. INTEGRATION OF MATCHING, INTERPOLATION, AND CONTOUR DETECTION

To use the surface smoothness model, it would be desirable to enforce truly three dimensional constraints rather than some partial implications of them. Before discussing how to do so, let us re-examine the source of three dimensional constraints, or, the model of three dimensional smoothness. Smoothness usually means that objects in a scene have faces which are smooth in the sense that the surface normal varies slowly. The faces meet at surface ridges which are themselves smooth curves in three dimensional space. This smoothness of the faces and ridges implies that the three dimensional boundary of any object against the background is also (piecewise) smooth. Thus, the image of a scene has the following structure. It contains regions corresponding to object faces. The border of a region may be composed of two kinds of segments: ridge segments, corresponding to surface ridges across which the surface slope is discontinuous, and occlusion segments across which surface depth is discontinuous. In the interiors of these regions both surface depth and slope vary smoothly.

3.1 The Need for Integration

The goal of the matching process is to select pairs of corresponding points in the two images. Matching provides surface depth values at the locations of the matched features. The estimated depth values at these locations constrain the type of surface that will result after interpolation. Effectively, therefore, the matching process determines the final surfaces derived. However, the scene surfaces are expected to follow the smoothness model. To make sure that this expectation is met, it is desirable that the matching process selects correspondences so as to eventually yield surfaces that are a close fit to the desired model. The interpolation process thus should be involved in matching so as to make acceptable matching decisions. The two processes should jointly and simultaneously determine the feature matches and the surface interpretation of the stereo data.

This suggests a *detect-interpret* approach in which the interpretation stage makes an integrated use of matching and interpolation processes. This approach is in contrast with the sequential *detect-match-interpolate* approach, where the matching decisions are made with only partial attention to surface smoothness, and the final interpolation stage must accept any suboptimal choices already made by the preceding stage of feature matching. We will say that the detect-interpret approach incorporates a *surface-smoothness* constraint to distinguish it from the partial constraints of *edge-connectivity* and *constant-local-disparity* discussed earlier.

3.1.1 Surface-smoothness versus constant-local-disparity

A demonstration was conducted to examine possible roles in human stereopsis of the *surface-smoothness* constraint and the *constant-local-disparity* constraint. A random dot stereogram was generated such that the use of the two different matching constraints would yield the perception of two different surfaces. The random dot stereogram portrays a surface whose height varies along the vertical axis as a cosine wave, and which is unambiguous everywhere except for a small region centered at the peak. This ambiguous region can be perceived as a smooth continuation of the cosine wave, or as a surface which is locally rough but has approximately constant height. The observer fixates at the depth midway between the two surfaces. Figure 3 shows the stereogram.

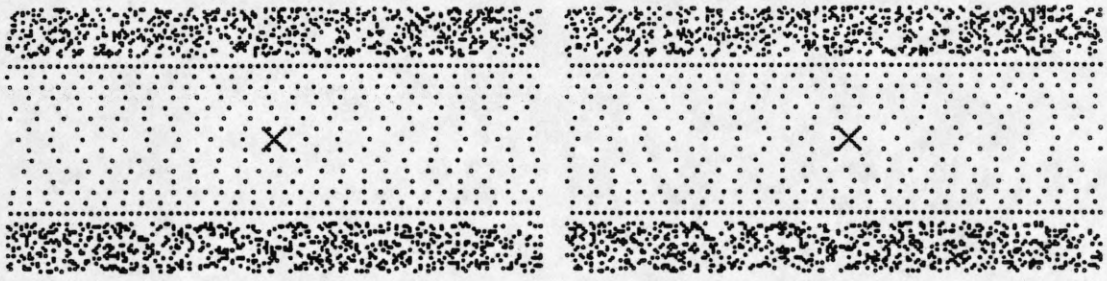

Figure 3. An ambiguous random dot stereogram portraying a cosine wave.

Viewing of the stereogram by human observers should lead to the perception of one of the two different surfaces, depending on the constraint used by the human visual system to resolve ambiguous matches. The smoothness constraint resolves ambigious matches such that the chosen matches define points that lie along a smooth surface in the three-dimensional space. The constant-local-disparity constraint resolves ambigious matches by choosing the match whose disparity has the same sign (positive or negative, convergent or divergent) as the majority of the points in a local neighborhood. This disparity is relative to the fixation point. It is assumed that the surface with the rough peak would be perceived if the constant-local-disparity constraint is used by humans, and the smooth cosine wave would be perceived if the surface-smoothness constraint is used.

Experiment

The equation of the cosine surface used in the demonstration was

$$z(x,y) = A \cos ky \qquad (1)$$

where $A = 52$ and $k = \pi/200$. The spatial units for these parameters are in dot widths. The width of a dot for this particular stereogram is about 1/72 inch. The x-axis is horizontal and the y-axis is vertical. The origin is at the center of the image. The offset, or disparity, for each corresponding dot in the right image is given by the value of z. The size of each image of the stereogram is 200 units (horizontal) by 100 units (vertical). The stereogram is divided into 3 regions: a central ambiguous region, from rows -31 to +31, and two unambigious regions, above and below the ambiguous region.

The unambiguous regions were created by placing dots at random in the left image, with 20% density, and placing the corresponding dot in the right image, according to the disparity given by the cosine equation above. The dots in the ambiguous region were placed only on every 4th row, beginning with row -28 and continuing to row +28. Therefore, there are 15 ambiguous rows. On each of these rows in the left image the dots were placed at regular intervals of w units, where w is different for each row. The dots in the right image were placed acording to the disparity given by the cosine equation; however, the matches are ambiguous because the dots on a single row can be consistently matched for any disparity which is an integral multiple of w.

For example, one of the ambiguous surfaces is perceived when the dots in each row in the ambiguous region are matched to yield a disparity that is w less than the true disparity. The absolute disparity of each row of the correct surface is shown in Figure 4 along with the dot spacing w for that row. The disparities of the incorrect matches are also given, which are just the disparities of the correct surface less w. The values of w were chosen randomly, but with the constraint that the disparity of the alternate surface not be less than 40. A cross section of the correct surface is shown in Figure 5, along with the incorrect surface which is described above.

Row	Correct disparity	w	Incorrect disparity
-32	45.6	-	- (last unambiguous row)
-28	47.0	3	44.0
-24	48.3	7	41.3
-20	49.4	6	43.4
-16	50.4	9	41.4
-12	51.1	8	43.1
-8	51.6	10	41.6
-4	51.9	9	42.9
0	52.0	12	40.0

Figure 4. The disparities of one half of the ambiguous section of the random dot stereogram (rows 4 to 32 are symmetrical around row 0).

The observers were told to fixate at the depth of a mark on the stereogram, which was placed at an absolute disparity of 46. Thus, the entire unambiguous portion of the stereogram has a negative disparity relative to the fixation point. The incorrect interpretation of the ambiguous surface also has a negative relative disparity. In contrast, the correct interpretation of the ambiguous surface has a relative disparity which is entirely positive. Therefore, if the constant-local-disparity constraint is used to resolve ambiguities, *i.e.* if that match is chosen which has the same sign as the disparities of the nearby points, then the lower (rough) surface should be perceived because the closest unambiguous points have a negative disparity. However, if the surface-smoothness constraint is used, then the upper surface should be perceived because it is smoother, *e.g.*, it has smaller mean curvature than the other surface.

The stereogram was shown to a small group of subjects (about 6). The subjects were at a distance of D or more from the stereogram. This minimum distance is necessary to ensure that both alternative surfaces are within Panum's fusional area. The two surfaces are a maximum of 6 units from the fixation point. If a maximum disparity of 15' is allowed for fusion, then this corresponds to a minimum viewing distance D of 19 inches.

All subjects perceived the smooth surface much more readily than the rough surface. Some were not able to perceive the rough surface at all. These results appear to support the hypothesis that the surface smoothness constraint is used. However, the results are not conclusive because the number of subjects was small, and it was difficult to ensure that the subjects were actually fixating at the depth of the mark.

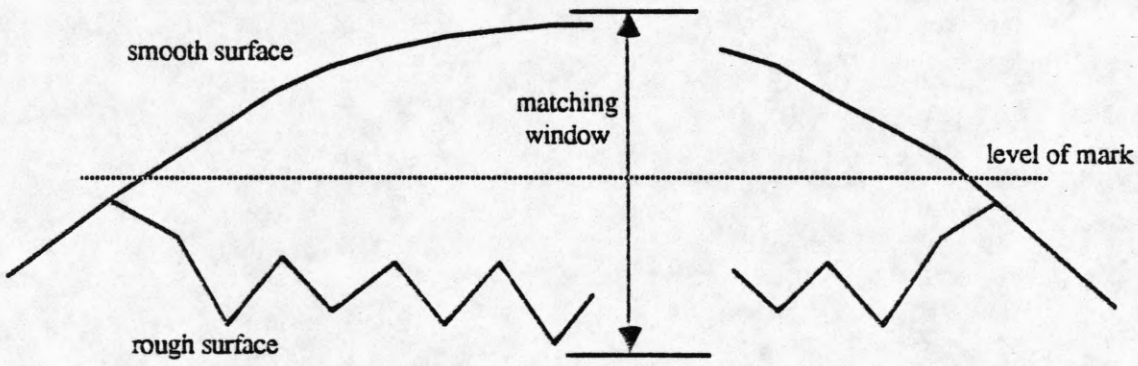

Figure 5. Cross section of ambiguous stereogram, showing cosine wave and rough surface.

It is interesting to note that using the figural continuity constraint instead of the constant-local-disparity constraint would not help. There are no long intensity contours in the random dot image, and so the figural continuity constraint would not be able to resolve the ambiguities. The surface smoothness constraint appears to be necessary to choose the correct surface.

3.2 An Integrated Approach

Each feature point P in, say, the left image may have a set of possible matches M_P in the right image, suggested by similarity of feature properties. Each candidate match in M_P determines a distinct point in three dimensional space that we will call a *depth point.* At most one of these candidate matches must be chosen from each set M_P (it is possible that no match can be chosen because the feature point is unmatchable due to noise or occlusion). The surface smoothness constraint is used to select a subset of depth points, at most one from each set M_P, such that nearby points lie on a smooth surface patch, with the different surface patches possibly separated by depth or slope discontinuities. The patches must contain as many points as possible. These patches thus represent local three dimensional consistencies among the depth points. Once a patch has been identified in the vicinity of an ambiguous feature point, the candidate match which is most compatible with the patch can be selected. This effectively integrates the processes of matching and surface interpolation.

Interpolation is necessary to fit the surface patches and implement the surface smoothness constraint. However, interpolation should not be performed across any surface discontinuities, and the locations of these discontinuities are not known until some interpolation has been done. Our solution to resolve this circularity problem is to fit the surface patches at each point in the left image, such that the patches are as large as possible. These patches will be large in size away from occluding and ridge boundaries, and small near these boundaries. From the spatial distribution of patch sizes, and a comparison between depths and slopes of adjacent patches, contours are identified in the image where no patches can be fit, and across which the depth or slope changes. Because of the local nature of patch fitting, the contours are expected to be only an approximation of the true ridge and occluding contours. The contours are located so that they are smooth and they well separate objects, or object faces. This effectively integrates the processes of surface interpolation and contour detection. We first described this integrated approach in [Hoff85]. See also [East 85] for a similar treatment of the stereo problem.

Once the contours are identified, they partition the original set of locally maximal patches into subsets, such that each subset covers a part of a smooth surface. The partitioning of patches also partitions the set of unambiguous points in three dimensions into subsets, each of which lies along a smooth surface. A global interpolation of a surface over each subset of depth points can then be safely performed. Alternatively, the global surface containing the subset of depth points can be estimated from a weighted combination of the heights of the locally maximal patches that contain the points in the subset, thus avoiding the computational expense of interpolation over large numbers of points. The latter approach was used in the implementation. The final result of the fusion process is a surface map in which ridge and occluding boundaries are explicitly specified, and they surround smooth surfaces.

3.3 Advantages of Integration

The main advantage of integration is the implementation of the surface smoothness constraint. There are additional advantages in using integration; some of these are described below.

3.3.1 Multiscale features and coarse-to-fine processing

The edge features may be detected at different scales, similar to the multichannel processing characteristic of human vision. The explicit surface representation provides the common ground for interaction among different channels. The sparsely located, coarse features can first be processed to estimate a low resolution approximation of the surface map. This surface map can predict the locations of matches of finer resolution features, leading to a finer resolution refinement of the coarse map. The current algorithm, described in the next section, does perform such coarse-to-fine processing. As discussed by Marr and Poggio [Marr79], coarse-to-fine processing also defines a role for eye vergence movements; namely, the current, coarse surface estimate can control the eye vergence so as to bring the left and right images into registration for the next, finer level of processing.

The primary reason for using coarse-to-fine processing in the current algorithm is efficiency. However, there are situations where the coarse level processing can succeed in fusing the images, but the finer levels cannot. Some of these situations include:

(1) One of the images is severely defocused.

(2) There is a large amount of uncorrelated high frequency noise.

(3) The two images are displaced vertically, by a small amount.

It has been shown by Julesz [Jule71] that human beings can fuse images with the above characteristics. Some of the images shown in Section 5 have these characteristics, and the results show that the coarse levels were able to fuse the images, but the finer levels were not.

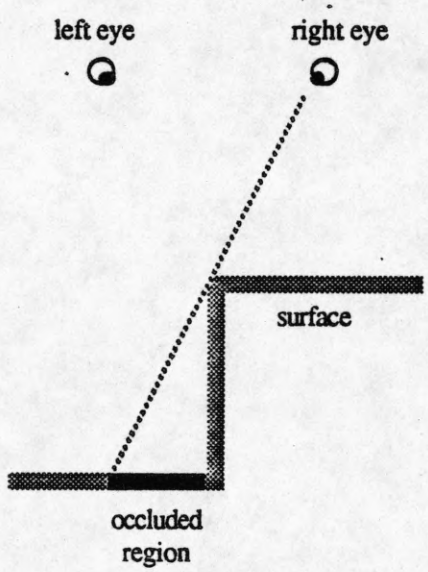

Figure 6. The black region is not visible to the right eye and constitutes an occluded region. If the relative heights of the two surfaces are known, the width of the occluded region can be calculated.

3.3.2 Occluded regions

When viewing scenes in which occlusion occurs, part of an occluded surface in the vicinity of occlusion border is not visible to one of the two cameras. Feature points in an occluded region have no correct matches, and it is important to identify occluded regions so that points within them will not be matched incorrectly. Given the current surface estimate, such occluded regions can be located (see Figure 6); therefore, the matches of any features in

one image whose matches are not visible in the other image are not used. The surface in such regions, devoid of stereo cues, may be assumed to be an extension of the estimated part of the occluded surface.

3.3.3 Transparent surfaces

It is possible that the depth points lie on transparent surfaces, one behind the other [Praz85]. For example, a scene could be viewed behind a dirty plane of glass. The features due to different transparent surfaces would be spatially intermixed in the image, and a two dimensional, local matching rule will not suffice. Since the surface fitting process only requires that the patches should have significant support from the depth points, it can yield multiple patches at any image location, one for each existing transparent surface at that location. Because the depth points are separated in three dimensions, they can be identified as belonging to different surfaces.

3.3.4 Disparity range independent of feature scale

The expected location of the match of a given feature is determined by the surface estimate available from the previous level. The disparity range over which matches are sought is completely adjustable, and does not have to be related to the parameters of the feature detector, *e.g.*, $\nabla^2 G$, as done in the Marr-Poggio approach [Marr79] and as pointed out by Mayhew and Frisby [Mayh80]. In principle, the range could be equal to the size of the image. This is because the number of false targets is irrelevant to the matching and surface fitting process; rather, the process relies on the existence of a set of depth points that define a smooth surface. As described in the next section, there is a constant amount of work required for each false target. The disparity range is reduced for practical reasons, to reduce the number of false targets.

3.3.5 Perspective shift in edge orientation

The candidate matches for an edge feature point are selected based on similarity of orientation. Typically, the orientations of two matched edges are expected to be the same. However, a surface edge when projected onto the two stereo images should have different image orientations, determined by the true three dimensional orientation of the surface edge and the camera geometry. Because of integration, the expected perspective shift in edge orientation can be taken into account by accepting those matches of an edge feature that not only yield depth points along a smooth surface patch, but also whose orientations are consistent with the orientation of the surface patch and the camera geometry. In other words, the matched edge features in the left and right images should have orientation differences not close to zero but to the values that would be expected if the local surface patch containing the given edge was viewed. Such perspective shifts often assume high values. Taking orientation shift into account would not be possible if surface information were not available at the time of matching. The implementation described in the next section takes such orientation shift into account.

3.3.6 Perspective compression of matching window

A surface region projected onto the left image will in general have a different image area than the same region projected onto the right image. If the left camera has a more frontal view of the surface, then the surface in the right image will be smaller, or compressed. Features will be more densely located in the right image than the left image; or, if the compression in the right image results in a merging of features, there will be fewer features in the right image than in the left image. In the latter case, there will be more unmatchable points in the left image than if the projected surface region was not compressed in the right image. Since, a coarse surface estimate is available (from the previous level), the expected increase in the number of unmatchable points can be estimated.

This is important to know because the number of unmatchable points is used as a criterion for whether a surface patch is a good fit to a set of depth points, as will be described in more detail in the next section. In the case of transparent surfaces, a different criterion may be required.

3.3.7 Continuity of discontinuities

Since the ridge and occlusion contours are explicitly detected, they can be constrained to be smooth because their counterparts in three dimensions are assumed to be smooth. Thus, the contours should be detected from the local patches such that they do not only locate discontinuities in surface depth and orientation but are also smooth themselves. This latter constraint is very useful because the feature locations in the image are usually sparse, particularly near a boundary of a steep surface.. The contours could locally move in the inter-feature (or "no-information") space without becoming inconsistent with the surface patches. The contour smoothness constraint makes it possible to propagate information between different parts of the boundaries to appropriately select the location of the contour when the local evidence does not lead to an unambiguous choice, or when it suggests a location that results in a large curvature. This should help in reducing the usually large number of depth errors that occur near surface boundaries. This step is implemented in the current implementation described in the next section, although only a coarse test of contour smoothness is used.

3.3.8 Focus of attention and computational efficiency

The availability of explicit surface and boundary information at any given level makes it possible to change the focus of attention at the next finer level of processing. Thus, the algorithm may not spend a large amount of computation at the next finer level in processing an area which is relatively featureless. Rather, it may concentrate on areas near object borders, in order to more precisely locate them. The explicit knowledge of border locations may serve to guide the processing at finer levels, thus allowing a surface representation to be computed in a shorter time. This may relate well with the savings observed in fusion time in humans for scenes containing depth discontinuities, as reported by Gillam et. al. [Gill84]. The current implementation does not incorporate the focus of attention, but instead processes the finer levels completely, regardless of the results of the coarser levels.

4. ALGORITHM AND IMPLEMENTATION

The outline of an algorithm to implement the integration approach is given in Figure 7. The algorithm works in a coarse-to-fine mode, and obtains depth maps at multiple resolutions. A given coarse level surface predicts the locations of edge matches at the next finer level. The matched features at the finer level provide a more refined surface which in turn predicts pairs of edges to be matched at the next finer level of resolution.

More specifically, the algorithm starts with an initial coarse estimate of the surface map, *e.g.*, a flat frontal surface at some depth. At each resolution level, then, the following steps are performed. First, edges are detected in each of the two stereo images. Matches are sought for the edges in locations predicted by the surface depth at the previous, coarser resolution level. Each possible match obtained corresponds to a point whose position in the scene as well as height are known. The match is recorded as a depth point in a (x,y,z) array by locating points with appropriate height z for each edge point (x,y). This results in a sparse set of spikes with tips that must lie on the surface. Second, smooth patches are fit to the depth points centered at each (x,y) position on a regular grid in the image. Third, a comparison of adjacent patches identifies those pairs that differ in depth or orientation. Such pairs of patches yield estimates of depth and orientation contours in an image. Finally, a smooth surface is interpolated within each region bounded by a closed contour to yield a piecewise smooth surface map at the given resolution. The process is then repeated at finer resolution using the current surface to predict matching locations of edges at the finer resolution. Processing at successively finer resolutions yields surfaces at increasingly fine resolution.

The process of fitting surface patches is used to break the circular interdependence of matching and interpolation, and integrate these processes. Since the patches are smooth and are a good fit to the data, they do not contain any contours, and therefore it is safe to use them for interpolation and disambiguation. The coarse-to-fine processing allows the integration of matching and surface interpolation across different levels of resolution.

The algorithm matches individual points in the left image with the corresponding points in the right image. Currently, these points are the zero crossings of the Laplacian of a Gaussian operator ($\nabla^2 G$). For each pair of corresponding points, the depth may be calculated from the disparity in the positions of the two points. Matching is driven from left to right, and from right to left, in two separate but identical processes. The result is that there are two sets of feature points, one for the left image, and one for the the right image. Each feature point is labeled with one or more disparity values.

In two identical and almost completely separate processes, two surface maps are constructed: one based on the coordinate system of the left camera, and the other based on the coordinate system of the right camera. The reason for this is that an occluding contour is easier to detect and locate from the viewpoint in which there is no occluded region. An occluded region from one viewpoint is the region next to the occluding contour which is not visible from the other viewpoint and thus is unmatchable. Therefore, each process detects only those occluding contours which are not next to any occluded region from that viewpoint. The contours detected from the two viewpoints are then combined, to give a complete set of contours. The surface map from either viewpoint can be displayed as the final result; in this work, the result from the left image is displayed. In the description that follows, the processing is performed identically for both the left-based and the right-based data, unless otherwise indicated.

The feature points have the following characteristics: First, they may have ambiguous depth values. Second, some of the points may have no correct depth value, due to noise or occlusion, or due to incorrect guidance for the matching from the coarser levels. Third, the feature points are sparse, which is characteristic of zero crossings. Fourth, the depth values are noisy, which can be caused by image noise, or the blurring effect of the $\nabla^2 G$ operator. In general, the uncertainty in the position of the zero crossing increases with the size of the operator.

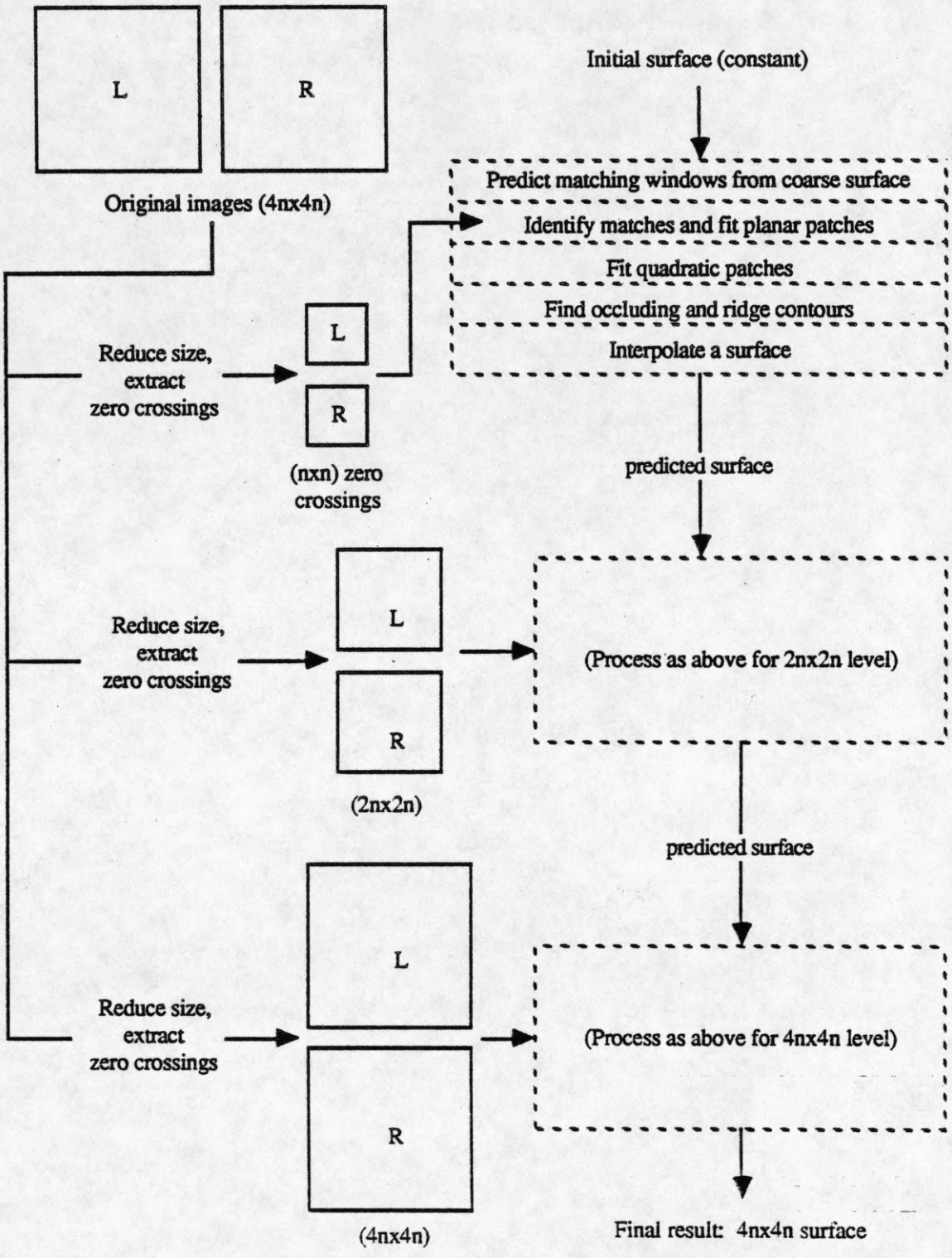

Figure 7. Control flow of algorithm integrating matching, interpolation and contour detection.

Smooth patches are fit to the depth points corresponding to features in each image. The algorithm first fits planar patches at the grid locations to find a rough approximation to the surface locally, and determine which combinations of possibly ambiguous matches are mutually consistent. To the depth points defined by these matches

then are fit quadratic surface patches to obtain a more accurate estimate of the surface. Depth and orientation contours are found by fitting bipartite planar patches, and detecting discontinuities between the two halves. At this stage, the contours found from the left image are combined from the contours from the right image. Currently, the algorithm uses the disparity values of the depth points instead of their depth values; the two, of course, are inversely related.

In summary, the processing steps for each resolution level are initial matching, fitting planar patches, fitting quadratic patches, detecting contours, and surface interpolation. The following sections give details of the current implementation.

4.1 Features Detected

The left and right images are each convolved with the $\nabla^2 G$ operator of the size (width of the Gaussian) corresponding to this resolution level. Zero crossings are then detected. The result is a pair of edge images. The effective width of the $\nabla^2 G$ operator (the diameter of the central negative region) was the same for each level, *i.e.*, 6.

The edges detected may be displaced from the true edge positions. Berzins [Berz84] did some analysis on the displacement error for specific image situations, and found that the error was usually much less than σ_G, where σ_G is the standard deviation of the Gaussian. Even in unusually bad cases, such as near very sharp corners or very small regions, the error was comparable to σ_G. We shall assume that for typical images the displacement error is normally distributed, and that 95% of the time the error will be less than σ_G. This is consistent with the displacement of zero crossings observed in our experiments.

In addition to the location, another characteristic of crossings that is used for matching is their orientation in the image plane. An edge segment on a surface has an orientation in three dimensions. For each zero crossing in the left image we can calculate the expected orientation of its match in the right image (see Figure 8) using the estimate of the surface orientation from the previous level, and search for candidate right image matches having the expected orientation, or occurring in the vicinity of the predicted location. Experimentally, however, we have found that the orientations of the right image zero crossings are not very close to those predicted by the orientations of the left image zero crossings and the camera geometry. Before proceeding further, we briefly describe the experiment.

Experiment

A plane with a synthetic random dot texture was projected onto the left and right image planes. The disparity gradient of the plane was 0.278 in the vertical direction and zero in the horizontal direction. The width of the $\nabla^2 G$ operator applied to each 256x256 image was 6. The maximum disparity difference between the endpoints of an edge segment occurs when the edge segment is oriented vertically, and the minimum disparity difference (*i.e.*, zero) occurs when the edge segment is oriented horizontally. Figure 9 shows the expected orientation change for zero crossing segments in the left image with orientations in the range from 0° to 180° (the orientation changes for the range 180° to 360° are the same as for 0° to 180°). Also shown are the average orientation changes observed from the matched pairs of zero crossings. The actual orientation changes are not close to the values predicted by the known planar orientation and the camera geometry, and the standard deviations are quite large (typically 6° or more). The experiment was repeated with planes at different orientations, and these results also showed that the actual orientation changes were not close to the predicted values.

The reason for the discrepancy between actual and predicted orientation changes is that the zero crossings are not located at the true edge position, but are displaced if the image intensities are locally nonlinear, or if the edge is not straight, *etc.* In general, the displacement from the true position of the left image zero crossing and that of the right image zero crossing are different, because the local image structure is different due to differences in perspective

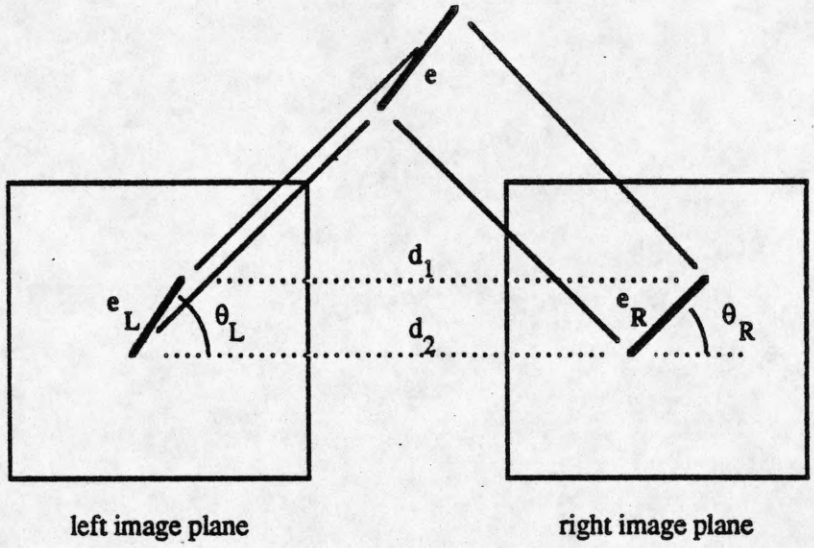

Figure 8. Edge segment e projects to segment e_L on the left image plane. If the three dimensional position and orientation of e is known, then the disparities of the endpoints of segment e_L, d_L and d_R, may be calculated. This fixes the position of segment e_R in the right image plane.

compression, noise, *etc.* Therefore, the shape of the zero crossing contour is distorted from the left to the right image in ways other than predicted by perspective difference, and so the orientations between corresponding pieces of a contour may be different than predicted. We therefore allow candidate zero crossing matches to have a large difference in orientation around the predicted value. In the implementation, a discrepancy of ±35° is allowed.

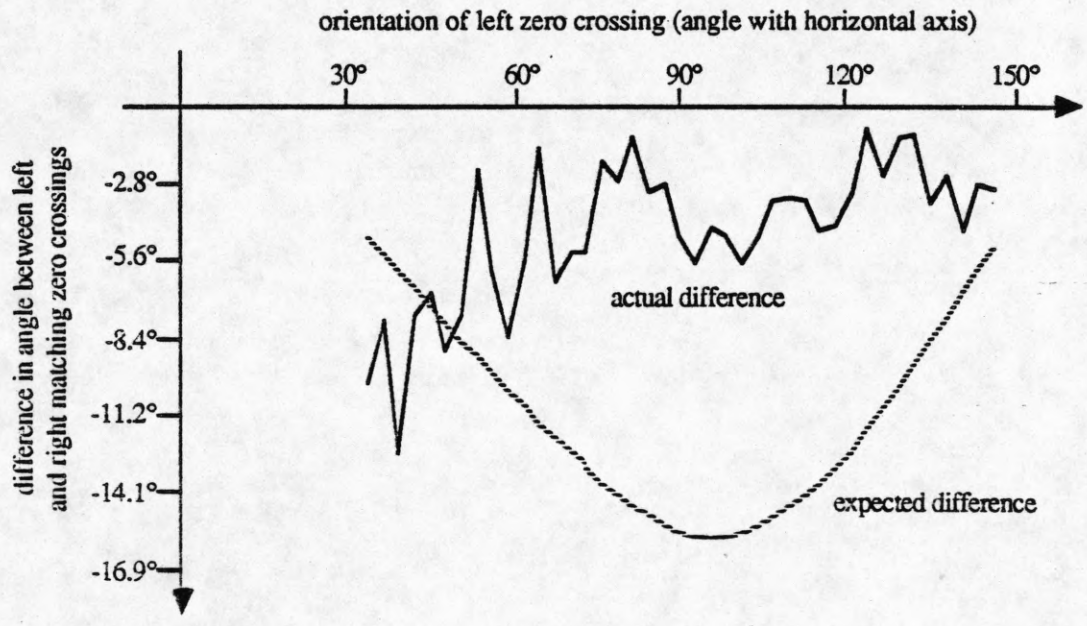

Figure 9. Expected and actual average orientation differences of zero crossings between left and right image, for an inclined plane with a random dot texture.

Figure 10 shows a stereo image pair that will be used to help explain some of the details of the algorithm. The stereo image pair shows a baseball resting on a newspaper. The size of each image is 256x256. The disparity ranges from about 7 pixels at the level of the newspaper to about 20 pixels at the top of the baseball. Because of the aspect ratio of the cameras, the spherical baseball appears to be an ellipsoid. Figure 11 shows the zero crossings extracted from the left and right images at 3 levels of resolution: 64x64, 128x128, and 256x256.

5.2 Surface Fitting and Contour Detection

In order to treat quantitatively the smoothness of surfaces and contours, models of surfaces and contours are needed.

5.2.1 Surface and Contour Models

The current implementation uses planes and quadratic surfaces as local models of real world surfaces. These surfaces are "smooth" in the sense that they are of low order. The depth of a point located at (x,y) according to the planar and quadratic models, $z_{p(x,y)}$ and $z_{q(x,y)}$, respectively, are given by

$$z_{p(x,y)} = a_1x + b_1y + c_1 \tag{2}$$

and

$$z_{q(x,y)} = a_2x^2 + b_2y^2 + c_2xy + d_2x + e_2y + f_2 \tag{3}$$

These models can approximate any continuous surface, if the regions or patches of approximation are small enough. In practice, however, these regions must be large enough to contain enough data points so that a reliable fit can be estimated. Thus, the surface reconstruction may be inaccurate for surfaces that are of higher order than quadratic. The amount of the error depends on the magnitude of the higher order coefficients of the surface, and the size of the patches used.

These models break down for regions that are discontinuous. In the vicinity of the discontinuity, a sizable planar or quadratic patch will not be a good fit to the depth points. Only patches whose size is very small (comparable to the average two dimensional spacing between depth points) will have a good fit. Two error measures are used to determine if the surface model fits a set of points: (1) the squared error of the points to the fitted surface, and (2) the number of points in the region which are unmatchable (for opaque surfaces).

The first measure depends on an *a priori* estimate of the noise in the depth values. Since we assume that the surface is locally planar or quadratic, the major component of this noise is due to the fluctuation of the zero crossings about the true edge position, due to image noise and the blurring effect of the ∇^2G operator. As discussed earlier, we assume that the displacement error is normally distributed, and that 95% of the time the error will be less than σ_G, the standard deviation of the Gaussian. Therefore, $2\sigma_N = \sigma_G$, where σ_N is the standard deviation of the noise.

Incorrect matches imply false depth points that are not on the surface. These points should not be allowed to affect the surface fit. In general, the error of fit of these points to the surface is large and does not have a normal distribution. To identify these false depth points, we assume that points with a displacement error greater than $2\sigma_N$ are due to incorrect matches. The distance $2\sigma_N$ is called the *outlier distance*. Points with displacements greater than this distance are probably due to matching errors and are ignored for the purpose of calculating the surface parameters. For a given number of points in a region, within the outlier distance to the surface, the probability of the sum of squared errors being less than or equal to ε^2 may be determined from the χ^2 distribution. The algorithm determines the maximum expected squared error for a 95% confidence level. If the squared error exceeds this value, the surface is rejected.

The second measure depends on an estimate of the probability of a point to be unmatchable. The surface patch should be rejected if there are too many unmatchable points. If the probabilities of the various points to be

unmatchable are independent, then the number of unmatchable points follows the binomial distribution. If the number of unmatchable points in a region exceeds the value for a 95% confidence level, the surface patch is rejected.

The probabilities of the various points to be unmatchable are *not* independent, because the zero crossings are not scattered randomly over the image, but lie on continuous contours. Short segments of zero crossing contours may be unmatchable, and so points which are near an unmatchable point and on the same contour are more likely to be unmatchable than points which are not. However, the assumption becomes less erroneous if the regions over which statistics are taken are large. We will assume that the probabilities are independent, for the purpose of analysis here.

The probability of a point to be unmatchable was determined empirically by projecting a plane with a synthetic random dot texture onto left and right image planes. The plane had a frontal orientation, *i.e.* was coplanar with the image planes. The only noise came from spatial and gray level quantization. The fraction of unmatchable points was about 10%, and so the probability of a point to be unmatchable was taken to be about 10%.

For a non-frontal planar surface, the probability of a point to be unmatchable increases with the magnitude of the disparity gradient of the surface, due to perspective compression of the matching window. The modified probabilities were also determined empirically, by counting the number of unmatchable points for planes of various disparity gradients. We found that the probability for a point to be unmatchable increases roughly as 0.2 times the vertical disparity gradient, and as 0.5 times the horizontal disparity gradient, if the horizontal disparity gradient is positive. The reason for the distinction between positive and negative horizontal disparity gradient is that in the former case, the apparent size of region decreases from left to right, so that some left points will have no right matches, whereas in the latter case, the apparent size increases from left to right, so that all the left points should have matches in the right image. Since the algorithm always has a current estimate of surface at any image location, the current estimate of the surface slope, *i.e.*, disparity gradient, is used to select the appropriate probability of no match from the empirically computed values.

To find occluding contours, we use the model of a bipartite surface patch: a circular region with two independent smooth (planar) halves, separated by a depth discontinuity at the center. The approach is analogous to that described by Leclerc and Zucker [Lecl84] for finding discontinuities in image intensities: it is necessary to find the local structure of the image (or surface) about the discontinuity in order to locate the discontinuity accurately. Our approach differs in that a fixed threshold is used to signal a discontinuity, instead of a statistical test. To find ridge contours, the model is the same, except that it uses an orientation discontinuity instead of a depth discontinuity.

The algorithm uses a coarse-to-fine mode of processing. The finest resolution is that of the original image. For the images in this paper, it is either 512x512 or 256x256. The coarser resolution images are obtained by convolving the original images with $\nabla^2 G$ filters of successively larger size, each time increasing the filter size by a factor of 2, and then subsampling the convolved images such that their size is reduced by a factor of 2. This continues until the coarsest allowed resolution of 64x64 is obtained. The steps described in the following subsections are repeated at each resolution level, starting at the coarsest level.

5.2.2 Fitting Planar Patches

We assume that the camera model is known, so that epipolar lines can be computed. The current implementation assumes horizontal epipolar lines, corresponding to parallel image planes. Searching for candidate matches is restricted to one dimension. The algorithm attempts to match only non-horizontal zero crossings, since the disparity of horizontal zero crossings is subject to large error. In our experiments, a zero crossing at an angle of 22° or less from the horizontal axis was classified as horizontal.

Figure 12 shows the non-horizontal zero crossings in the left baseball image, at the 64x64 level of resolution. The orientation of each zero crossing is shown as a number from 0 to 127, corresponding to the range of angles 0° to 360° from the positive x axis. The algorithm ignores zero crossings that are close to an image border (less than 1.5w) because their postions and orientations are usually distorted by the strong edge generated by the image border.

Planar patches are fit in circular image regions centered at each point along a regular grid. The spacing of this grid is w, the filter size. A sparse grid is used to reduce the amount of computation. At each grid point (i,j), the quadratic surface estimate $z_{i,j}(x,y)$ obtained from the previous level is used to match the zero crossings in the circular region. Up to two planes are then fit to the depth values obtained by matching the points. This is done for a sequence of radii, starting at a radius of w, up to a maximum of 2w. The largest possible disc is identified at each point under the constraint that the depth points in the disc are a good fit to a plane, using the two error measures described earlier.

Two planar fits are obtained instead of one, in order to delay the final choice until information from adjacent regions is available to reliably choose between the two. Also, if there are two surface estimates for this point from the previous level, the process is repeated for the second estimate, resulting in up to four planes for the region.

To ensure a reliable planar fit, the data points must be distributed over the entire region. This condition is tested by examining if the convex hull of the points on the image plane contains the region center; or equivalently, that the vectors to the points from the region center span an orientation range greater than 180°. If this condition is not satisfied, the plane solution is rejected.

A crucial part of this algorithm is the use of the Hough transform [Duda73] to fit planar patches. Identifying the best-fitting planar patches in the vicinity of an image point requires selecting the most planar subsets of depth points among all possible combinations of mismatched and ambiguous points. Using a standard least squared method such as Gaussian elimination would lead to combinatorial explosion, because a different plane would have to be fit to each possible subset of depth points in a region. The Hough transform is a relatively inexpensive and robust method of fitting planes having least squared error.

To implement the Hough transform, a three-dimensional parameter space is set up with each dimension corresponding to a parameter in the equation of the plane:

$$z = a\,x + b\,y + c \tag{4}$$

For each depth point (x_i,y_i,z_i) in a circular region, cells in quantized parameter space are incremented at the locations corresponding to the solutions (a,b,c) of the equation

$$c = z_i - ax_i - by_i + \varepsilon_i \tag{5}$$

where e_i represents the amount of error of fit of the point (x_i,y_i,z_i) to the plane represented by (a,b,c). The array is incremented at each location (a,b,c) by the amount ε_i^2: the squared error of fit of (x_i,y_i,z_i) to the plane (a,b,c). After all points have been considered in this manner, the minimum entry in the parameter array represents the solution with the minimum squared error.

If a point is further than the outlier distance (twice the estimated standard deviation of the noise) from the plane, it is considered to be an outlier to that plane, and its squared error does not contribute to the total. In the case of ambiguous points, only one of the depth values contributes to any plane: the one which is closest in depth. There are two kinds of ambiguities: two left feature points matching the same right point, and two right feature points matching the same left point. Both kinds of ambiguities are taken into account. Because of outliers and ambiguous points, the solution found is not the true least squared error solution, but the solution with the least squared error among the points satisfying the above conditions.

Currently, transparent surfaces are not allowed, because all the points are expected to be on the same surface, except for the relatively few mismatches. One way to allow for transparent surfaces would be to keep track

of the mismatched points that actually belonged to another valid plane, and not count them as mismatches for the purpose of rejecting the current plane.

An important advantage of the Hough transform is that it requires a constant amount of work for each depth value, and the amount of work is not exponential in the number of ambiguous points or mismatches, as would be the case with Gaussian elimination. A disadvantage of the Hough transform is the limited resolution of the parameter space. In the implementation, the parameter space was 7x7x16, with the first two dimensions used for the x and y slopes from -0.6 to 0.6, and the third dimension for the z offset. This allowed a resolution of 0.2 in the slope, and 1.0 in the z value. A higher resolution could be used, but at the cost of additional computation.

One way of circumventing this problem is to use a Hough array with adaptive resolution; the resolution is dynamically increased in the parts of the parameter space where peaks are found at the coarser level. However, because the planes obtained are only local approximations, a very fine resolution is not crucial. A maximum allowed slope of ±0.6 is used because the probability of a point to be unmatchable increases with slope (as described earlier), and planes having greater slopes than this would have so many unmatchable points that they could not be distinguished from planes fit to random matches.

Figure 13 shows the results of fitting planar patches to the baseball stereo image pair, at the coarsest (64x64) level of resolution. The planar patches are represented by circles, and are centered on a grid which has a spacing of w. The circles correspond to the actual radii of the planar patches.

4.2.3 Fitting Quadratic Patches

The planar patches fit in the previous step represent a rough approximation of the surface. Quadratic patches are now fit at each grid point, to the depth points which are now unambiguous. The quadratic patches may be fit over a larger region of the image than the planar patches, because they are of higher order and can better follow the surface curvature. The main purpose of the plane-fitting step is to determine which combinations of matches are mutually consistent, so that a quadratic surface may be fit to only those combinations.

To fit a quadratic surface centered at grid point (i,j), the following procedure is used. The planar patches centered at the neighbors of (i,j) are tested for mutual compatibility. Two neighboring planes are *compatible* if the depth and the orientation differences between them are less than certain thresholds. In the implementation, the depth difference threshold is w/2 and the orientation difference threshold is 0.25. Two incompatible planar patches at neighboring grid points are likely to be separated by an occluding or ridge contour, and the two patches should not be part of the same quadratic surface. (These contours are found in later processing.)

The planes in the neighborhood up to 2 grid points away from (i,j) are placed into sets, such that the members of each set are compatible with each other. This is done by the following procedure: For each plane in the neighborhood (in arbitrary order), the parameters of the plane are transformed so that it is centered at (i,j). The transformed parameters are then compared with the averaged paramters of each set. If it is compatible with the average of one of the sets, then it becomes a member of the set; else it becomes a member of a new set. The two sets with the most members are now chosen, and the rest are discarded.

For each set, the planes in the set should be local approximations of the same quadratic surface. Therefore, the matches consistent with these planes should lie on or near the quadratic surface. These matches are found by rematching the zero crossings, using the same method as described earlier, and keeping the closest alternative to the plane within the outlier distance. A least-squares quadratic surface is fit to these matches, using Gaussian elimination. As before, the squared error is compared to the maximum expected error as given by the χ^2 distribution, and the fit is rejected if the error is too large.

The quadratic surface containing the most points is kept as the fit for the grid point (i,j). The number of points is used as a criterion because reliability and accuracy increase with the number of points. In fact, it was found

that discarding quadratic surfaces with less than 30 points removes most of the incorrect patches, caused by a local excess of spurious matches.

Figure 14 shows the result of fitting quadratic patches for the baseball example, for the 64x64 level of resolution. The quadratic patches are centered at the locations of the star-like objects, and are on the grid of w spacing. The quadratic patches are not circular, but consist of a union of the circular planar patches in the neighborhood. The length of the lines of the stars represent the size of the quadratic patches in the given direction.

4.2.4 Locating Contours

The next step in the algorithm is to locate occluding and ridge contours. At such discontinuities, the quadratic patches are usually missing, or contain few points, or have a large and systematic error. This is because the quadratic model is not adequate to fit the data at those locations. One way to detect discontinuities is to search for quadratic patches that are missing or have large errors. However, this method is not reliable because the patches may be missing or defective for other reasons, such as image noise, or lack of image texture. Also, there may be patches that straddle contours, if the data points in the vicinity are sparse.

A more reliable method to detect discontinuities is to look for two adjacent surface patches that differ in depth or orientation. This method is implemented by fitting bipartite circular planar patches at each grid point in the image. The circular patches are divided into two halves by a diameter of a given orientation. A plane is fit independently to the depth points within each semicircular half of the bipartite patch, using the Gaussian elimination method for least squares. If the two planes differ in depth (or orientation) by more than a threshold, then there is evidence for an occluding (or ridge) edge in the vicinity of the grid point, and in the direction of the diameter used to obtain the two semicircles.

To obtain the depth points needed for the plane fitting, the zero crossings in the patch are matched using a disparity estimate obtained from the closest quadratic patch (or patches) to the point. If a depth point is ambiguous, the closest match to the surface (obtained from the quadratic patches) is used for the plane fitting. Only matches within the outlier distance to the estimated surface are retained. To help increase the reliability of the planar fit, the radius of each semicircle is increased until it contains at least 15 points, from a minimum radius of 3w to a maximum radius of 5w. The same two tests are used as before to decide if the planes are a good fit to the data points: (1) the binomial test, for the number of unmatchable points, and (2) the χ^2 test, for the total error of the points to the plane.

If the two planes are a good fit to the data points, and they differ in depth (or orientation) by more than a threshold, then there is evidence for an edge point in the vicinity of the grid point. The threshold used was w for depth, and 0.25 for slope. To locate the edge point more accurately, the following procedure is used: The bisector of the bipartite circular patch forms an edge segment that partitions the matches into two sets, belonging to one side or the other. This edge segment is moved to and fro between the two sides. As the edge segment moves, some data points that were originally in one side of the bipartite patch become members of the other side. At each position, a score is computed, which is the average squared error of fit of the points to the surface whose side they are currently in. The edge point is placed at the position with the lowest score. This method yields good localization of the edge point, while saving the expense of fitting the edge detector at every point.

For occluding edge points, the above procedure is modified by establishing a "dead zone" adjacent to the edge segment, on each side of the segment. Any matches inside the dead zone are ignored and they do not contribute to the score. The reason for this is that the zero crossings near occluding contours are typically distorted by the contour and the matches are not reliable. The distortion is typically significant out to a distance of w/3, and so this is the width of the dead zone that was used.

The bipartite circular edge detector is applied four times at each grid point to detect edges at the four orientations: vertical, horizontal, and the two diagonals. If an edge is detected at a particular orientation, it is localized and given a score, as described above. The edge orientation at this grid point with the best score is retained.

This edge detector may have multiple responses, *i.e.*, it may signal the presence of an edge at multiple locations in the vicinity of the true edge, along a line perpendicular to the true edge. However, the best score should ideally occur at the position of the true edge. Therefore, the edge points which are not a local minima in the direction perpendicular to their orientations are suppressed.

The result is a set of candidate edge points, lying near the centers of the bipartite planar patches where they were detected. These edge points were detected and localized based on local evidence. To accept an edge point we further require that it falls along a smooth (ridge or occluding) contour. This enforces the property that objects as well as their faces have smooth borders. This constraint is particularly useful since the edge points are often sparsely located and the local evidence may place the contours only approximately, somewhere in the interfeature area. The requirement of contour smoothness propagates the placement constraints on the edges posed by matched features occurring in different parts of the contour, thus, possibly resolving correctly the local uncertainity.

In the implementation, cubic splines are fit to the locally detected edge points to obtain a smooth contour. Cubic splines are continuous up to the second derivative, and are not required to pass through the given points [Fole83]. A more sophisticated method would integrate contour smoothness with contour detection. For example, a cost function could be defined such that one part consisted of the scores for the local placement of the edge point, and another part would minimize the local curvature of the contour, *i.e.*, the curvature of the curve connecting the edge point and its neighbors along the contour. The contour would be placed such that it gives a minimum value for the cost function, *i.e.*, it optimizes a combination of local placement scores and local smoothness.

An occluding contour is easier to detect and locate from the viewpoint where it is not adjacent to an occluded region, *i.e.*, a region which is not visible to the other viewpoint and thus is unmatchable. If points within an occluded region are matched, they match at random, and occasionally the matches define small, false surface patches. Whether an occluded region lies next to an object boundary depends upon the viewpoint. In the left image, there may be a region to the left of the left object boundary which is occluded from the right viewpoint. However, the right object boundary is not adjacent to any occluded region. In the right image the left boundary of the object has no adjacent occluded region but the left boundary does. Therefore, in two identical and almost completely separate processes, two surface maps are constructed: one based on the coordinate system of the left viewpoint, and the other based on the coordinate system of the right viewpoint. Each process detects only those segments of occluding contours which have an orientation such that there is no occluded region in that viewpoint. Matching is driven from left to right in one process, and from right to left in the other process. The result is that there are two sets of feature points, one for the left image, and one for the the right image. Each feature point is labeled with one or more disparity values. The contours detected in the other viewpoint are then combined with the contours detected in the current viewpoint, to give a complete set of contours. The surface map of either viewpoint can be used to display the final result; in this work, the result from the left image was used.

False ridge contours are occasionally detected parallel to occluding contours, on either side. These arise because surface patches are occasionally fit across the occluding edge, forming a steep ramp. To eliminate these false ridge contours, the algorithm eliminates ridge edge points near occluding contours. In addition, patches which overlap contours are eliminated, because they are adversely affected by the depth points on the other side of the contour, resulting in an incorrect surface estimate.

Figure 15 shows the detected edge points for the baseball example, at the 64x64 level of resolution. The edge points are all ridge edge types, and have orientations indicated by the short line segments attached to each edge point. Figure 16 shows the remaining edge points after suppressing non-minima in the directions perpendicular to

the orientations of the edge points, and the cubic spline that was fitted to the edge points. In this example, all points on the cubic spline contour are labeled with a "2", which is the code for a ridge edge. Other possible codes are: "1" for an occluding edge, and "3" for an occluded point.

4.2.5 Generating a Surface Map

The final step is to interpolate to obtain a complete depth map, and predict matching locations for features at the next finer level. The quadratic patches are a good local estimate of the surface, defined at the grid points. To interpolate the depth at each point P on the surface, the closest patch or patches to point P are used, which do not lie across any occluding or ridge contour. In the implementation, patches were used if their centers were up to 2w from point P. The computed height at point P is the average of the heights of individual patches weighted according to their distance to P. If there are no patches within 2w of P, than no attempt is made to interpolate a depth at P, and it is marked "unknown". If the point P is in an occluded region, no attempt is made to interpolate a depth, and it is marked "occluded". A reasonable guess for the depth at such a point can be made by extending the surface which is more distant out to P; this, however, was not done.

To predict matching locations for the next level, the set of quadratic patches is copied to a new grid, twice the size of the old. There are now new grid points which do not have a quadratic estimate, and these must be interpolated from the existing ones. In addition, in the vicinity of occluding contours, we would like to provide two depth estimates — one from the high side of the surface, and one from the low side. This is done because the location of the occluding contour is known only coarsely, and we would like to be able to match the zero crossings in the vicinity of the contour.

To accomplish the above, the following procedure is repeated N times, to propagate the known estimates to the unknown areas (N = 4 in our implementation). At each grid point on the new level for which there are less than two estimates, examine the 8 neighboring quadratic patches. If these patches are mutually compatible (meaning their parameters have similar values), then average them to determine the quadratic estimate at the new point. If they are not all compatible, divide them into compatible sets, and average the ones in each set to determine up to two estimates.

Figure 17 shows the surface interpolated for the baseball example, using the quadratic patches and edges obtained earlier, for the 64x64 level of resolution. For display purposes, the vertical scale is not equal to the horizontal scale, but is chosen so that the vertical range is about half the horizontal range. Locations where the surface is "unknown" are displayed at the lowest height; for example around the borders of the image. Finally, locations which are "occluded" (there are none in this figure) are displayed at a low height, but slightly above the height of "unknown" locations.

Figure 18 shows the quadratic patches and detected edges for the 128x128 level of resolution. Figure 19 shows the surfaces interpolated for that level. Figure 20 shows the quadratic patches, detected edges, and occluded regions. For the finest, 256x256 level, Figure 21 shows the corresponding surfaces. The baseball does not look spherical for two reasons: First, the aspect ratio of the cameras makes it look elongated. Second, the surface displayed is not a *depth* map, but is a *disparity* map. The unit of measure in the vertical direction is not length, but is the number of pixels of displacement between the left and right image.

To calculate the depth map from the disparity map, one must measure the positions and parameters of the cameras accurately. Since this was not done, it is not possible to test the results by physically measuring distances to points in the scene and comparing them to the output of the program. However, in this example it was possible to derive the true disparity map for the scene by measuring the disparity of selected points by hand, and using the equation of an ellipsoid for the baseball and the equation of a plane for the newspaper. The ideal surface is shown in

Figure 22. The difference between the calculated surface and the ideal surface is shown in Figure 23. Points which have an error of more than one pixel of disparity are shown in Figure 24.

Some of the errors along the occluding boundary have magnitude approximately equal to the height of the baseball from the table, indicating that they are due to a misplacement of the occluding boundary. Nearly all of the errors away from the occluding boundary are much smaller.

The next section illustrates more results obtained with the algorithm described above. The results are presented in the same format as the baseball example, but not in as much detail.

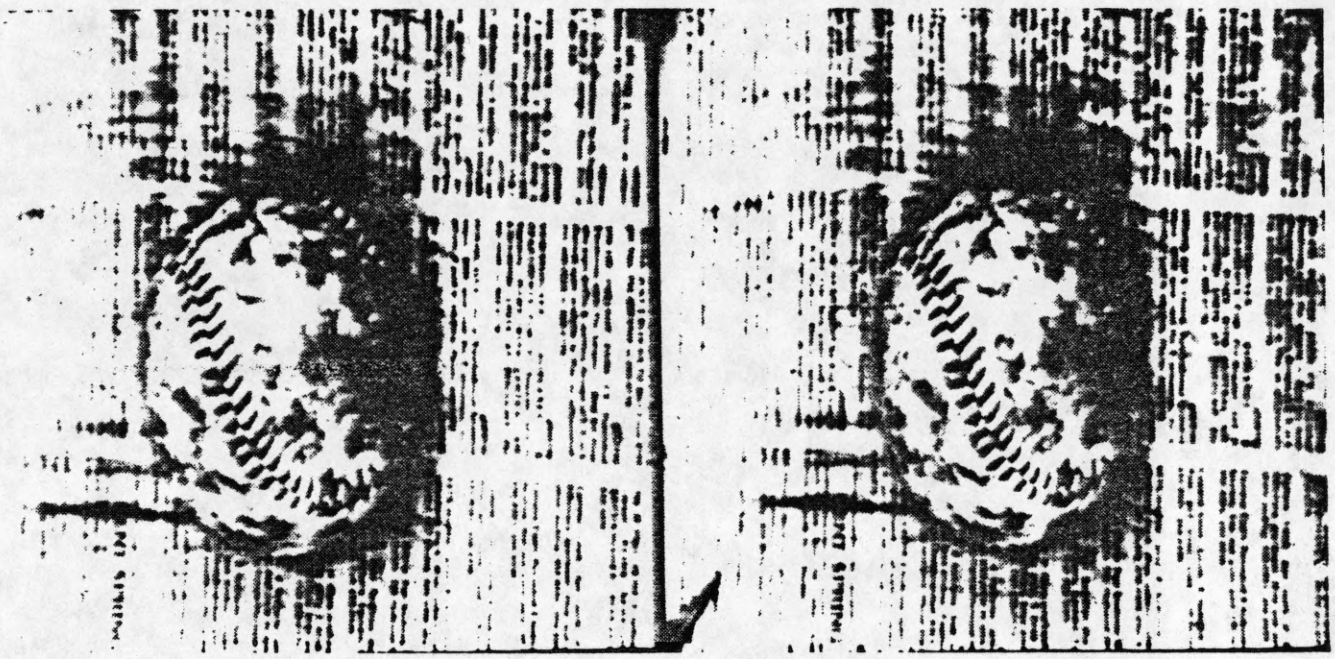

Figure 10. A 256x256 real image of a baseball on a newspaper. The dispaity ranges from about 7 pixels at the level of the newspaper to about 20 pixels at the top of the baseball.

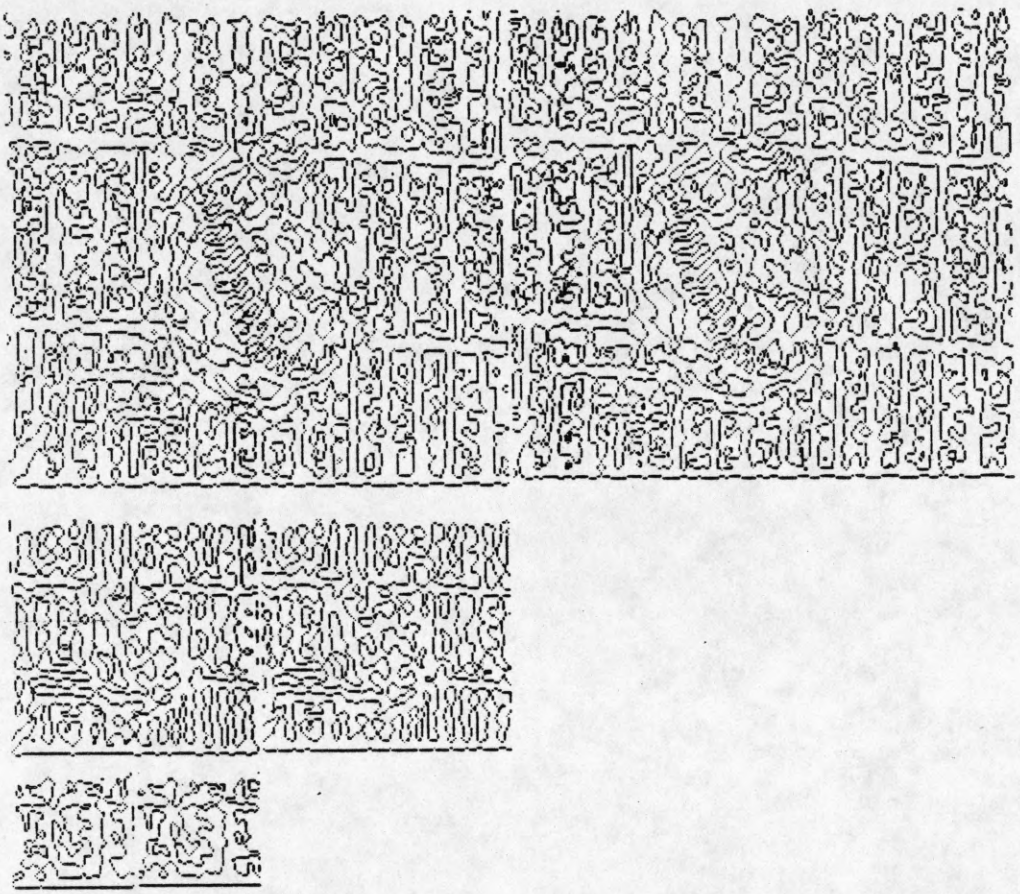

Figure 11. Zero crossings detected for the baseball image, at 3 levels of resolution: 64x64, 128x128, and 256x256.

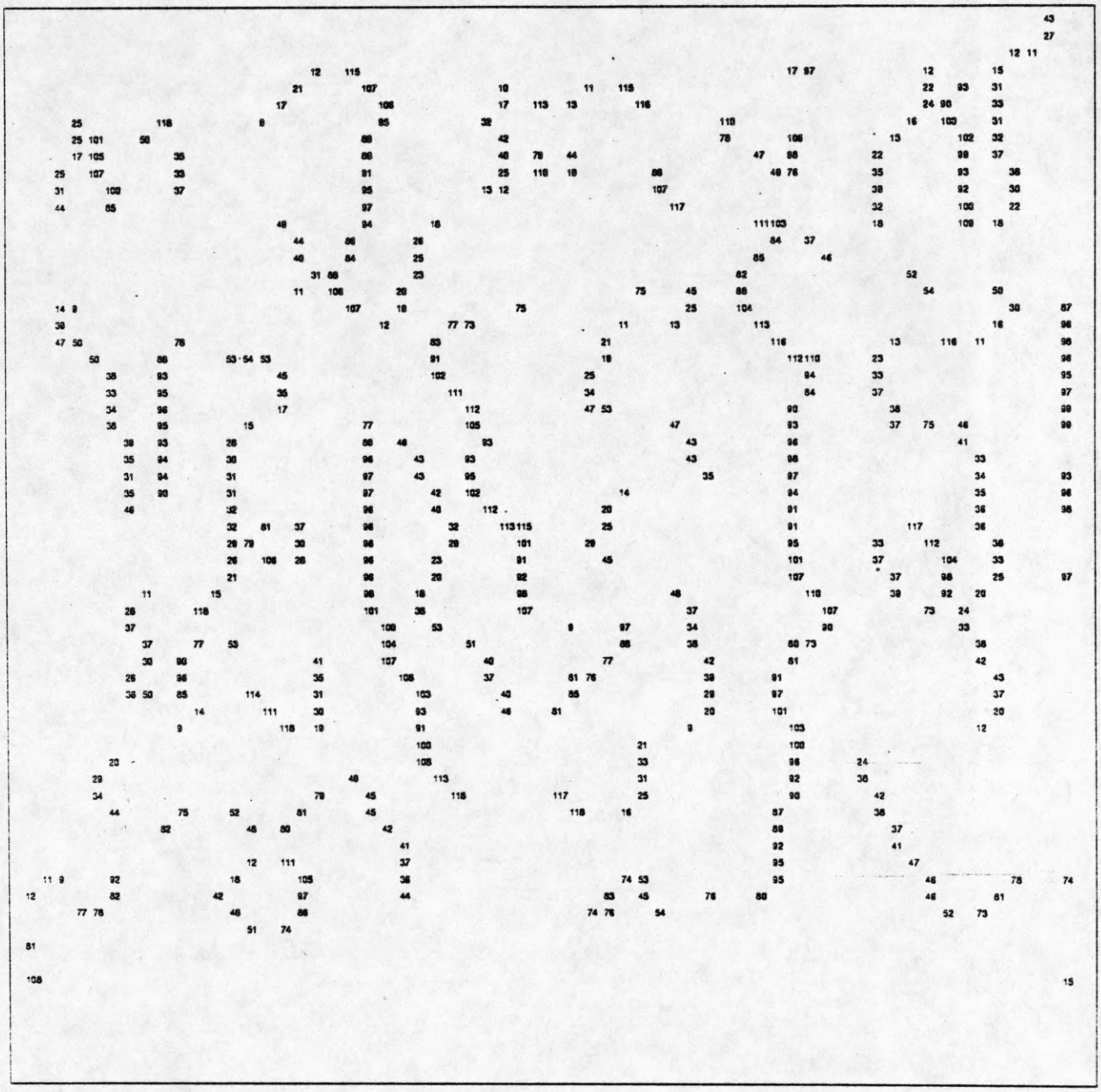

Figure 12. Non-horizontal zero crossings for the left baseball image, at the 64x64 level of resolution. Orientations 0..127 correspond to angles 0°..360° counterclockwise from the x axis.

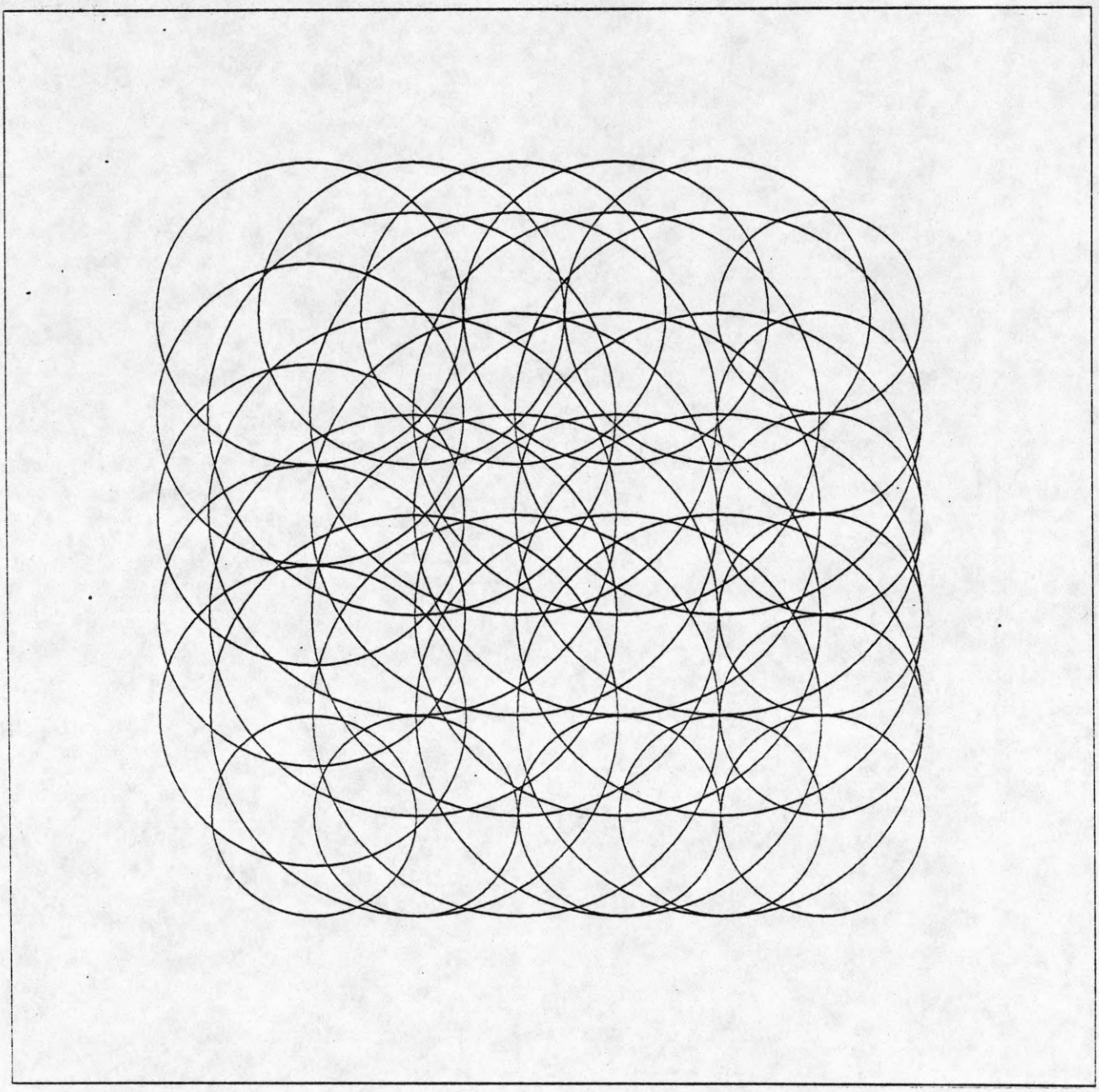

Figure 13. Planar patches found for the baseball image, at the 64x64 level of resolution. Patches are centered on a regular grid of whose spacing is 6 pixels.

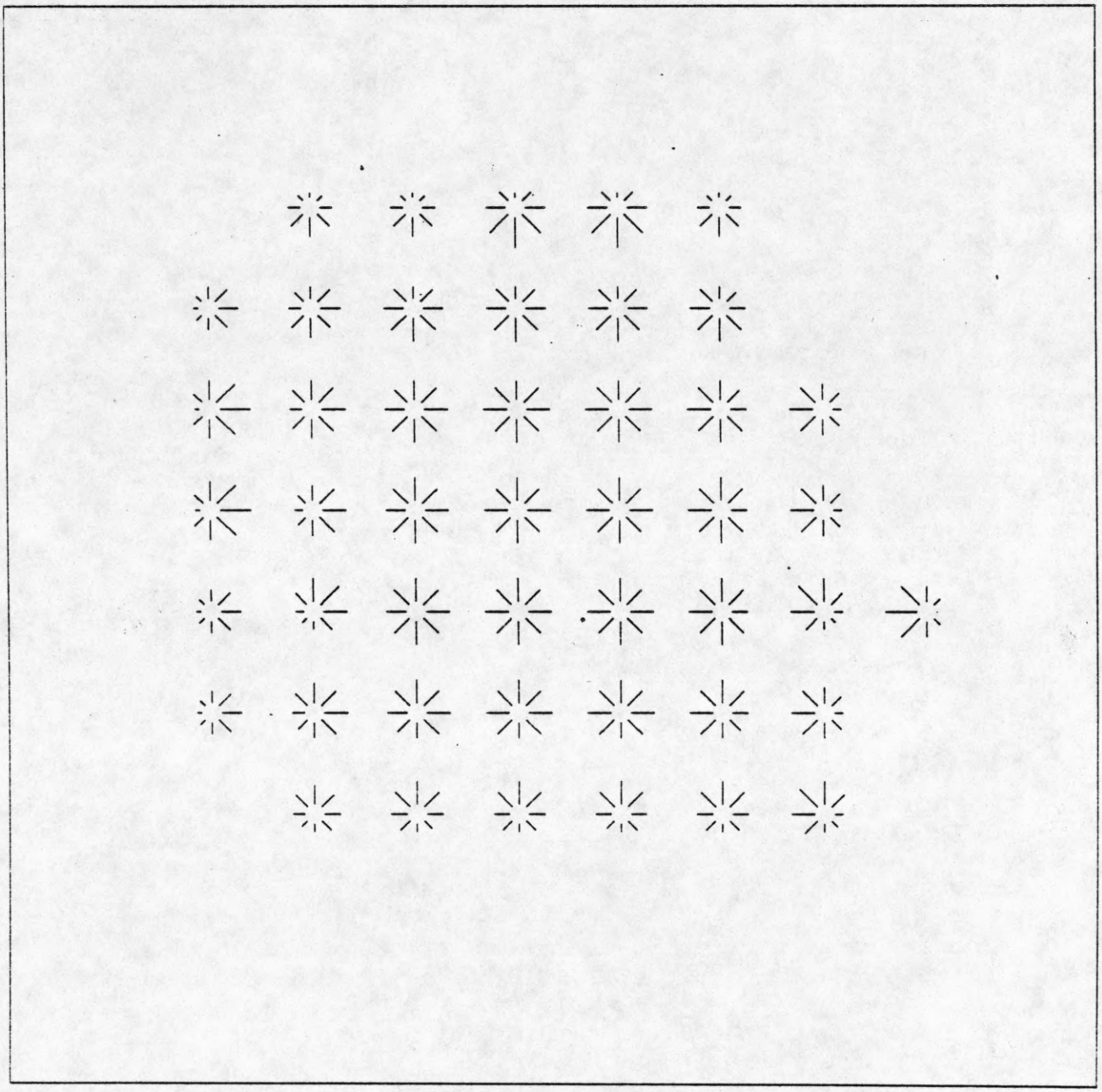

Figure 14. Quadratic patches found for the baseball image, at the 64x64 level of resolution.

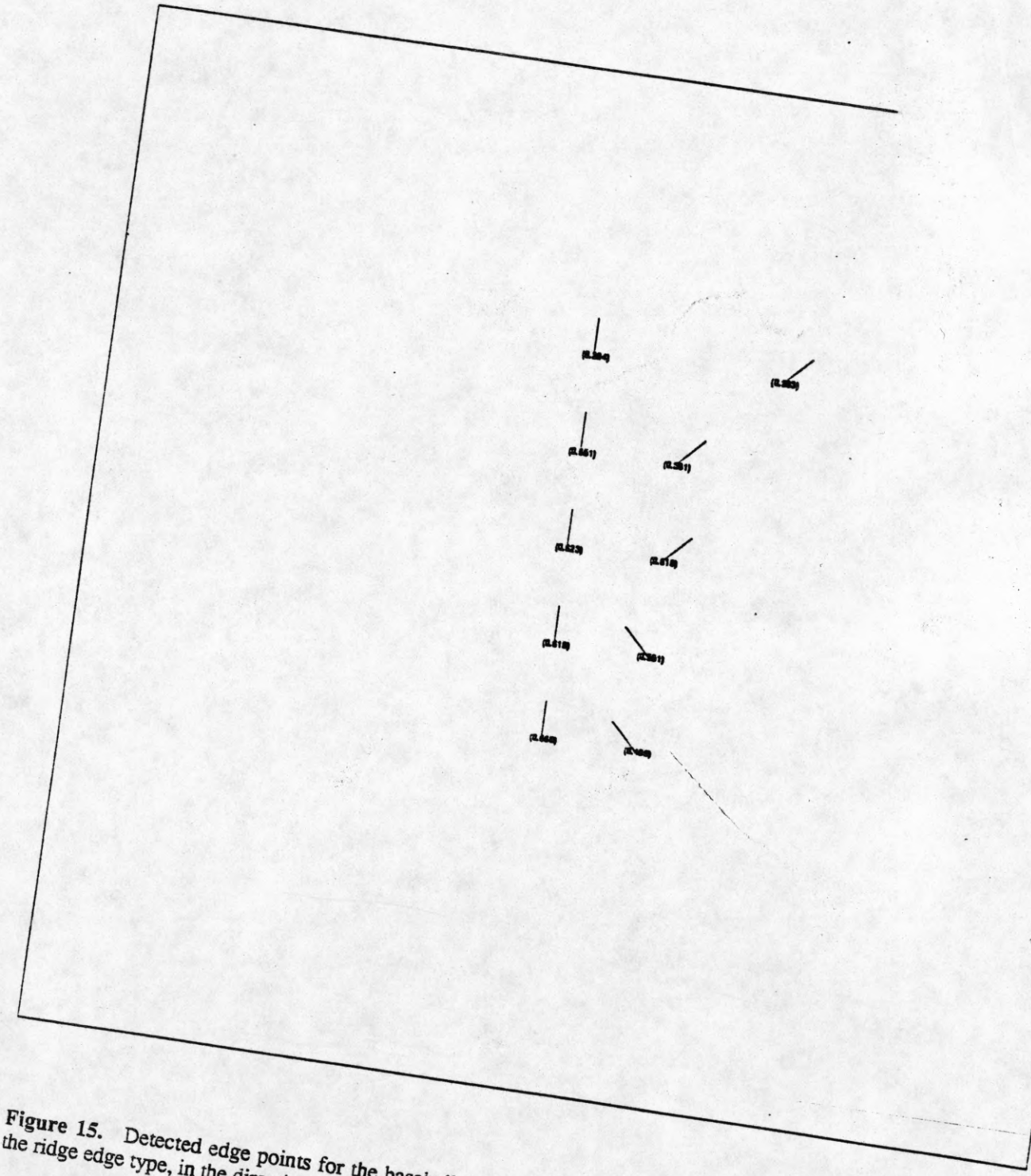

Figure 15. Detected edge points for the baseball image, at the 64x64 level of resolution. Edge points are all of the ridge edge type, in the directions shown.

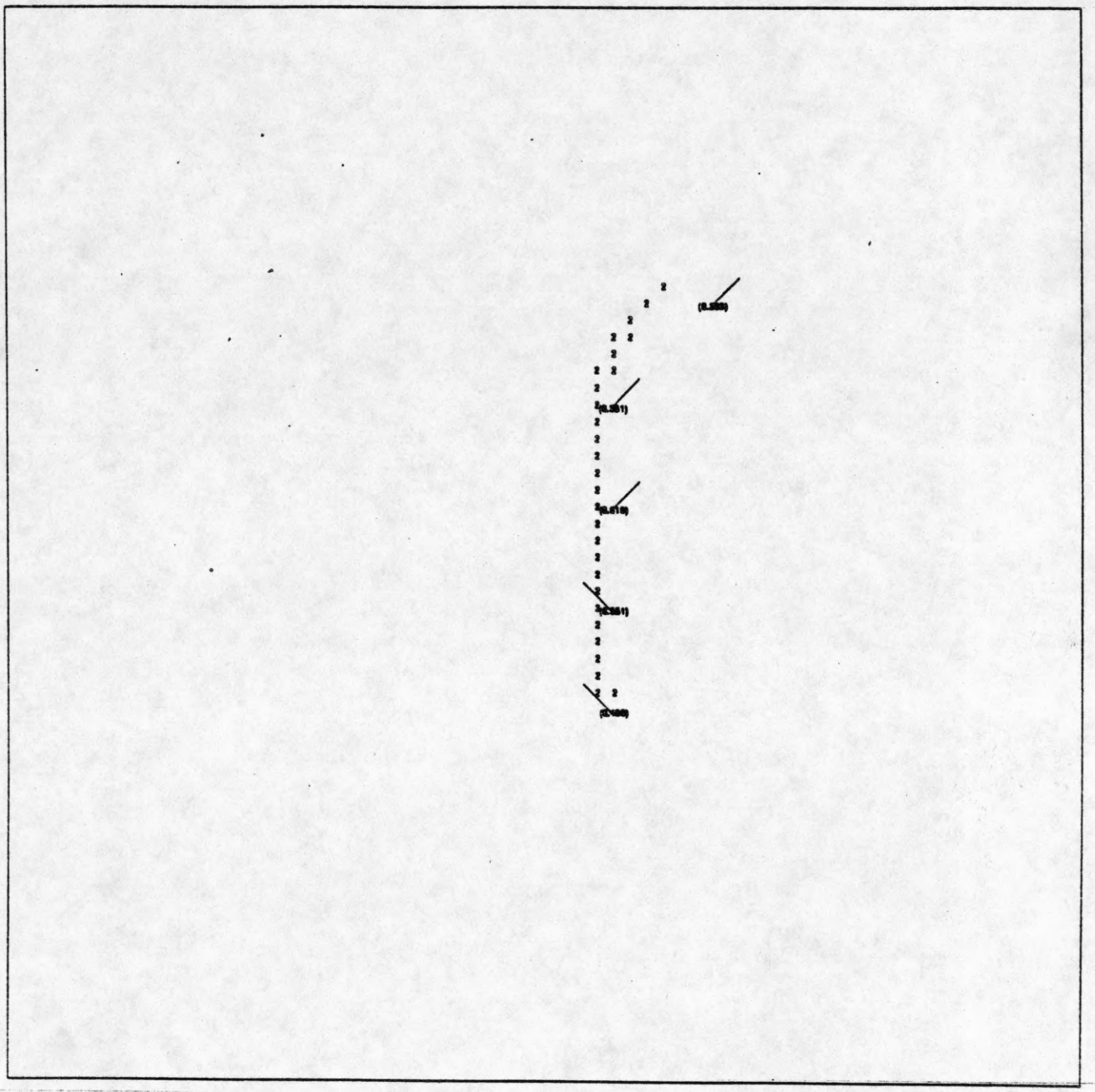

Figure 16. Edge points remaining after non-minima suppression for the baseball image, at the 64x64 level of resolution. A cubic spline contour, indicated by the "2"s, is fitted to these edge points.

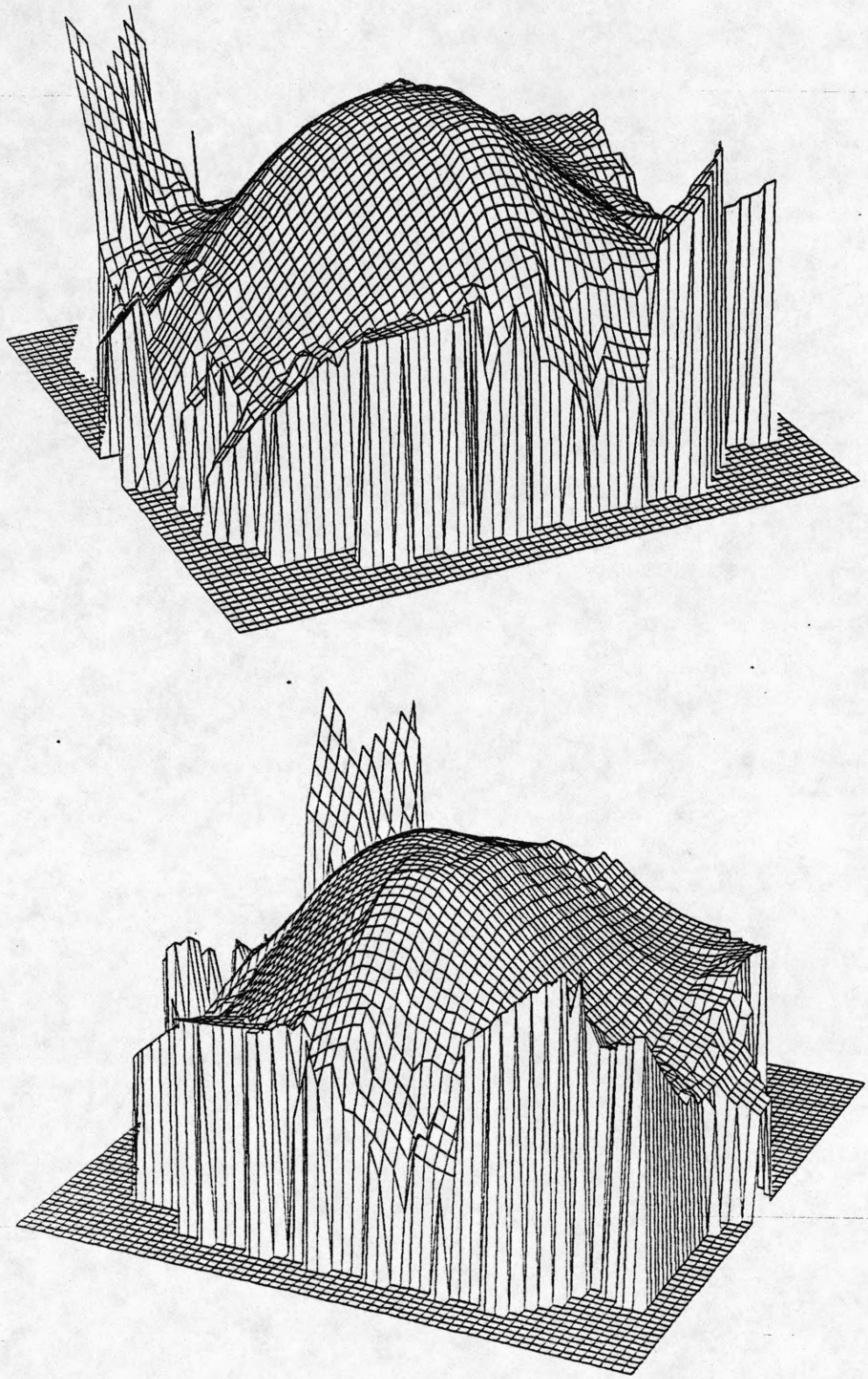

Figure 17. Reconstructed disparity surface for the baseball image, at the 64x64 level of resolution. Disparity ranges from -2 to 5 pixels.

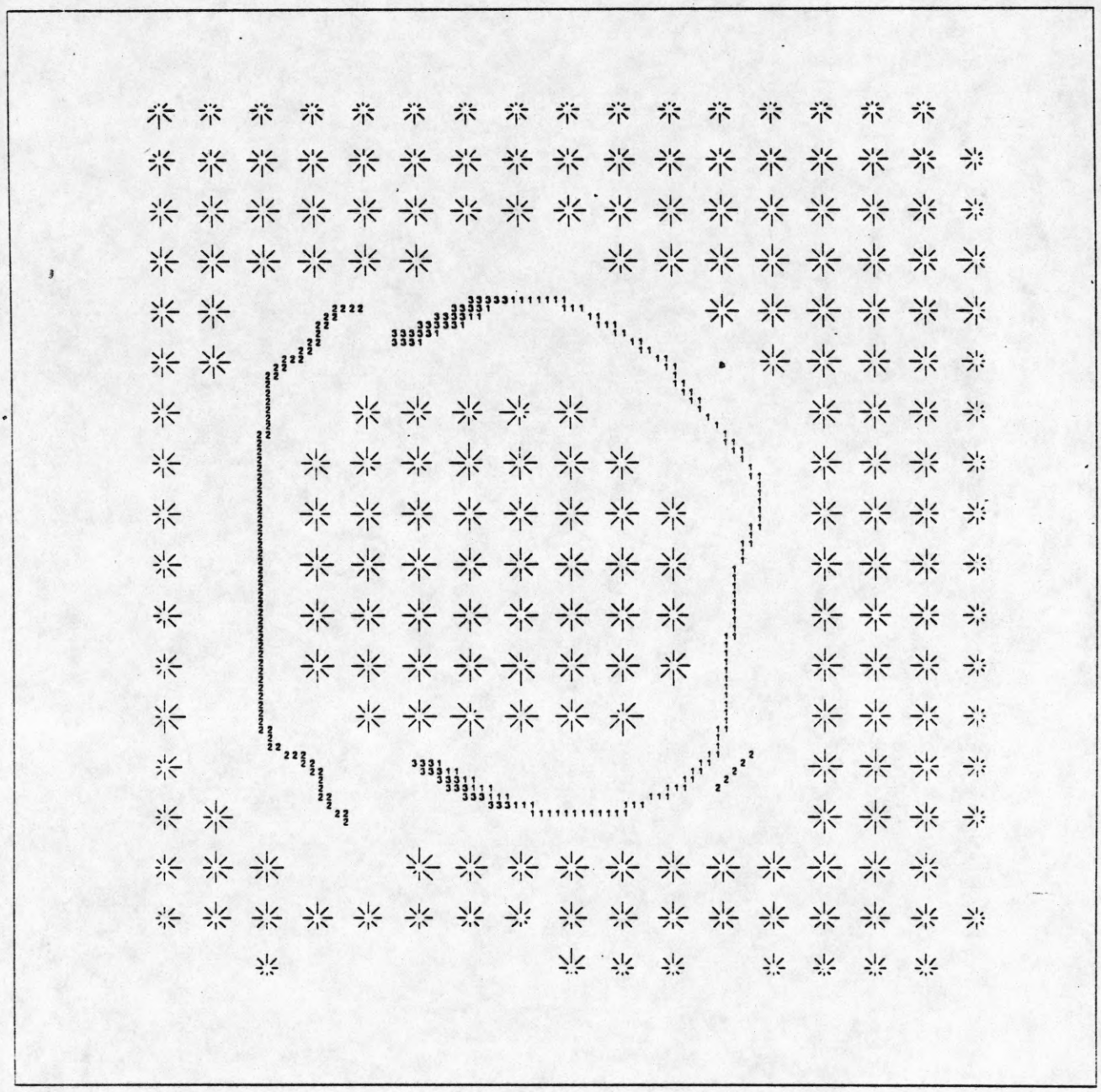

Figure 18. Quadratic patches and detected edges for the baseball image, at the 128x128 level of resolution.

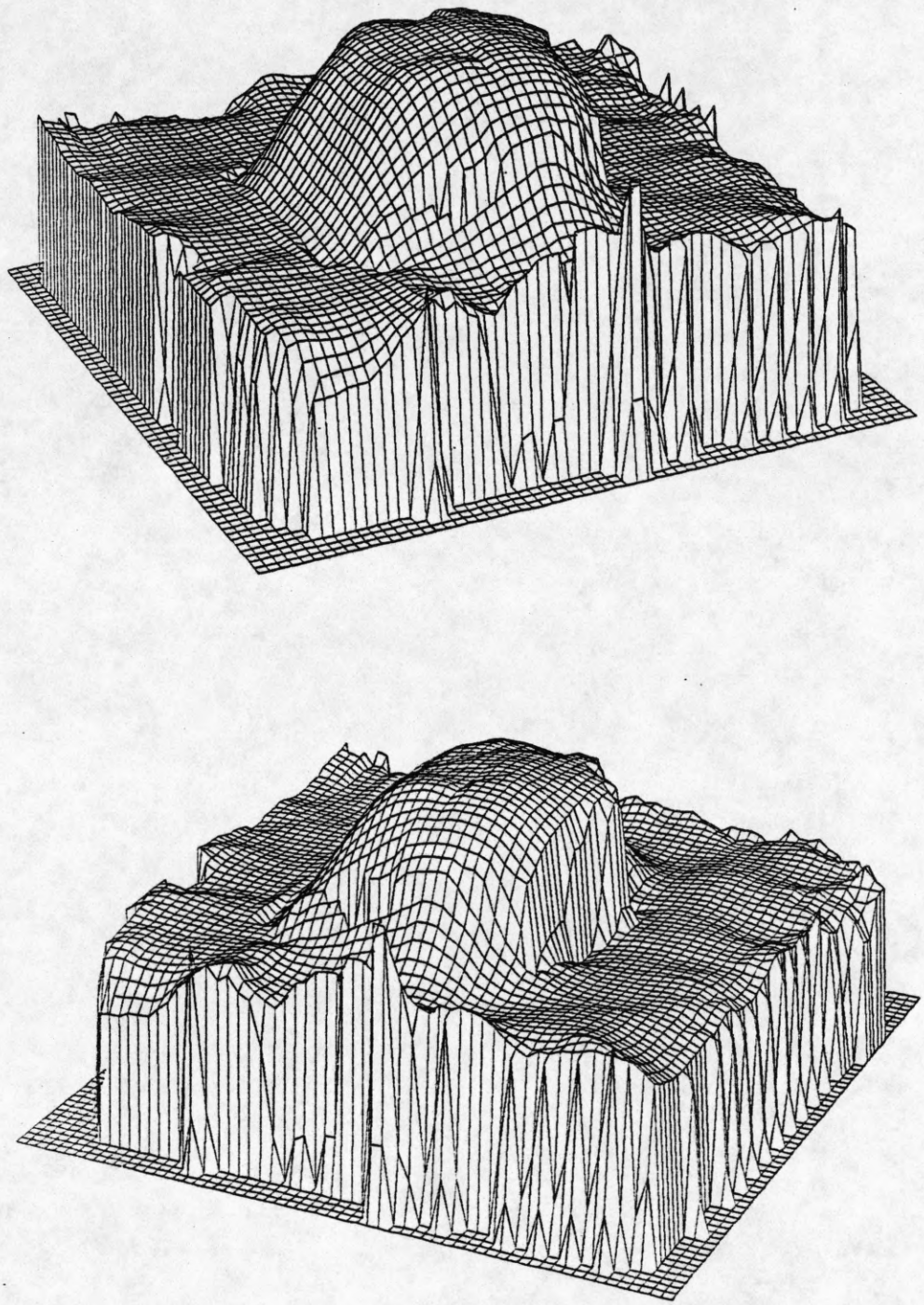

Figure 19. Reconstructed disparity surface for the baseball image, at the 128x128 level of resolution Disparity ranges from -2 to 10 pixels.

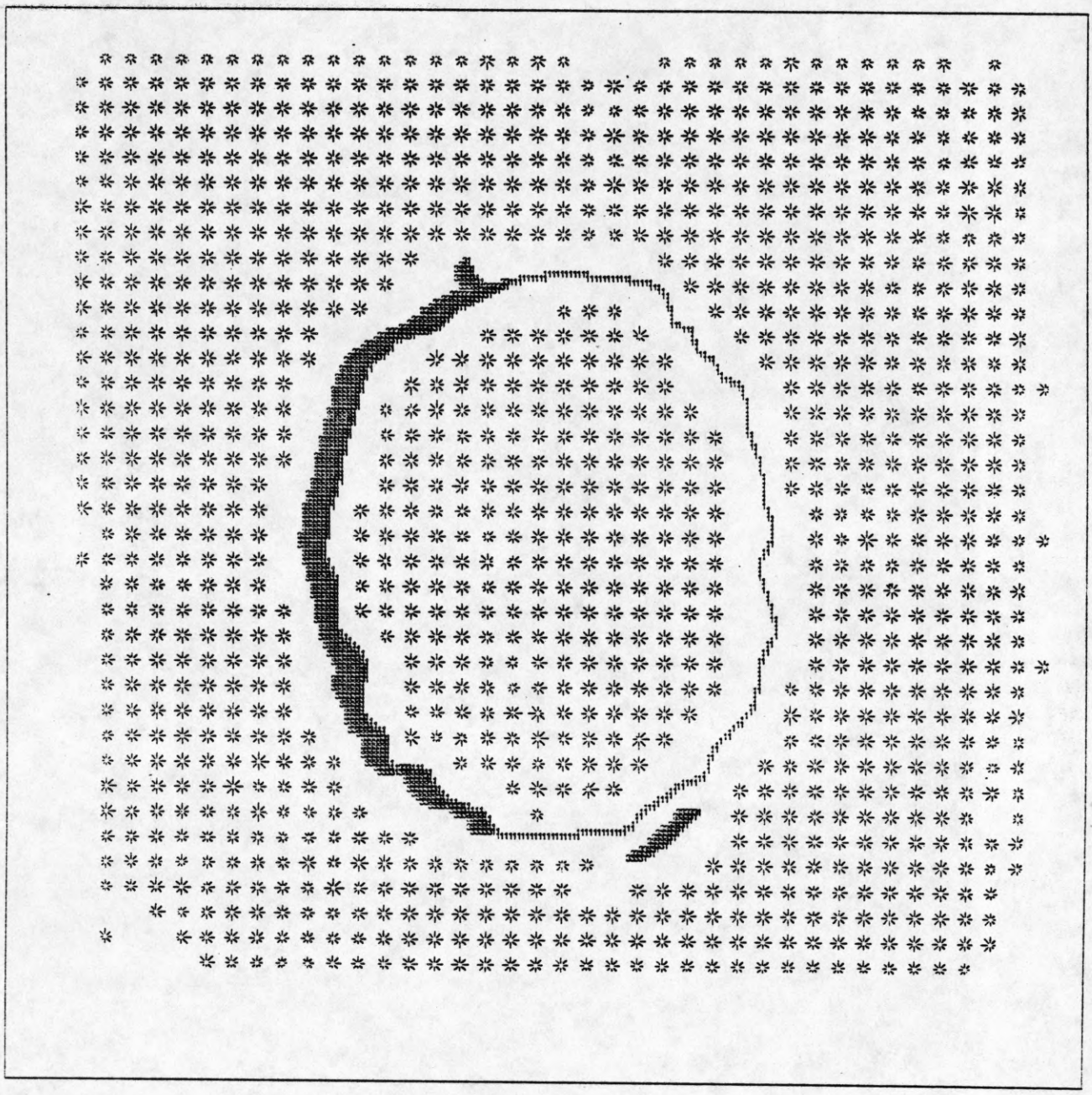

Figure 20. Quadratic patches and detected edges for the baseball image, at the 256x256 level of resolution.

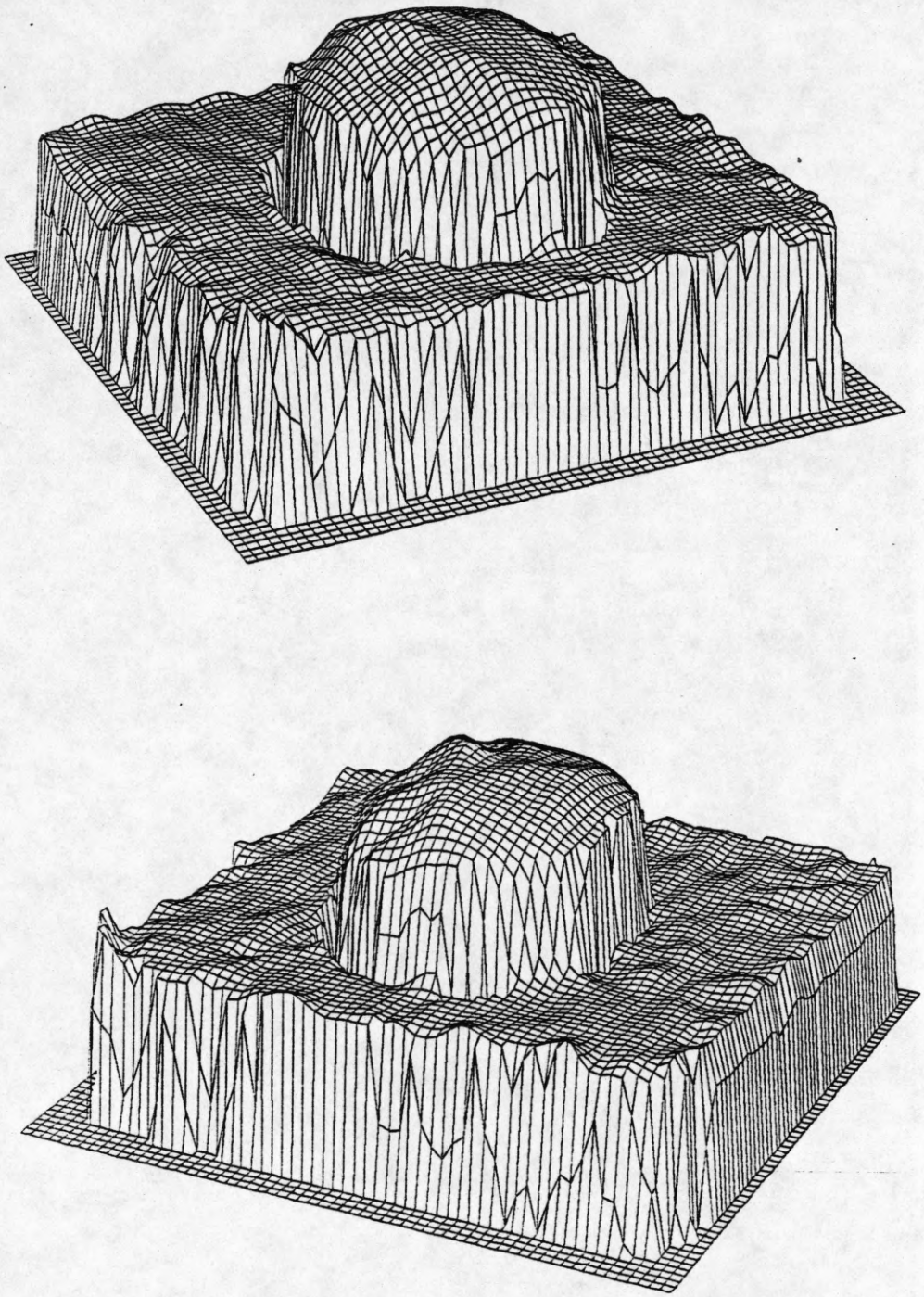

Figure 21. Reconstructed disparity surface for the baseball image, at the 256x256 level of resolution Disparity ranges from -2 to 20 pixels.

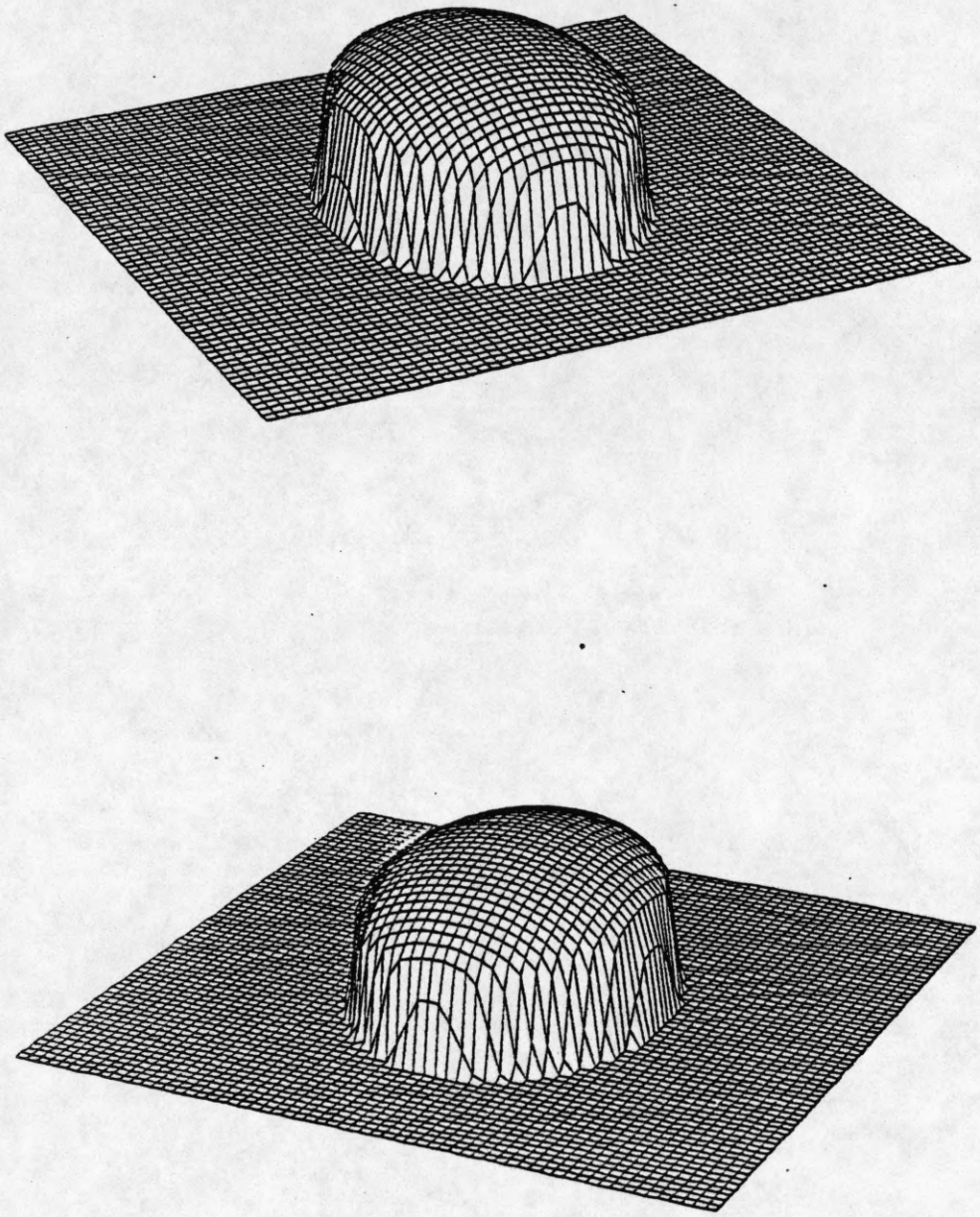

Figure 22. Ideal disparity surface for the baseball image. Disparity ranges from 7 to 19 pixels.

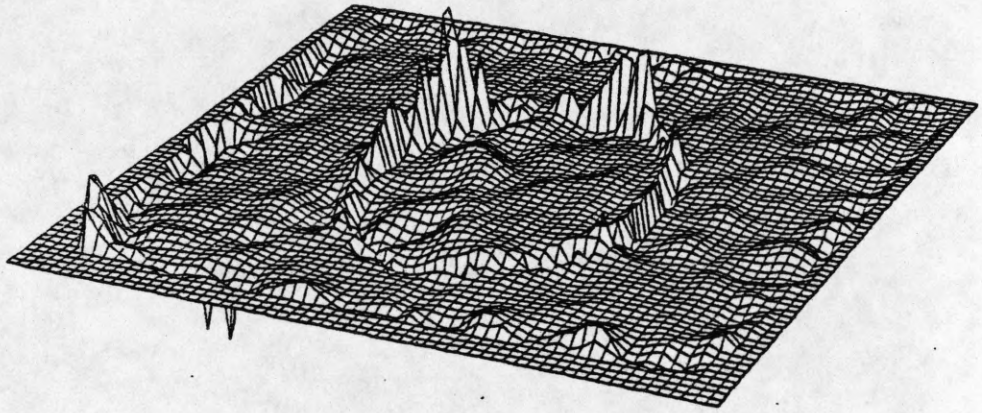

Figure 23. Difference between ideal and calculated disparity surfaces, for the 256x256 level. Disparity ranges from -14 to 7 pixels.

Figure 24. Points which have a disparity error greater in magnitude than one pixel, for the 256x256 level of resolution.

5. EXPERIMENTAL RESULTS

Results are presented for running the algorithm on a set of synthetic images and a set of real images. The synthetic images were generated by a program, which wraps image textures onto three dimensional surface patches, and displays the result in a perspective view. The synthetic images have the advantage that the exact ground truth is known, so the results can be quantitatively compared to the output of the stereo algorithm. For the real images, the ground truth was measured by hand at selected points, to confirm the accuracy of the results. The reconstructed surfaces were also qualitatively compared to the surfaces perceived by one of the authors.

Most of the real stereo images were taken using a single camera at two different positions. The positions and orientations of the cameras were approximately, but not precisely measured. Several example images were obtained from other laboratories. If the two view directions are parallel, then ideally the two images should be vertically registered; *i.e.*, the epipolar lines are horizontal. However, some of the images were not vertically registered, because the view directions were not exactly parallel, causing the epipolar lines to be non-horizontal. Although it is not difficult in principle to calculate the positions and orientations of the epipolar lines from the camera parameters and imaging geomtry, we found it easier to correct the images manually so that they were vertically registered. This was done by compressing or stretching one the images in the vertical direction until it was aligned with the other. This procedure was just a first approximation to the task of correctly registering the two images, and was done so that the program could be run on examples that it otherwise would not be able to handle. The procedure was not intended to completely correct for all non-ideal camera optics and viewing situations.

For each stereo pair, the size of the finest level of resolution was specified, along with a constant estimate of disparity to be used for the coarsest level. The finest levels were either 256x256 or 512x512, and the coarsest level was always 64x64. We manually measured the range of disparities of the stereo pair and chose the midpoint to obtain the constant estimate.

Image regions which have no significant intensity texture generate very weak edges. The zero crossings detected for such regions are due to noise for the most part, and are uncorrelated. A threshold was used to eliminate zero crossings due to weak edges, using the slope of the image convolved with $\nabla^2 G$ at the zero crossing. With no threshold, isolated patches were occasionally fit in these regions. These were incorrect, and appeared to be due to local groups of zero crossings happening to lie on a consistent surface, by chance. The same threshold was used for most of the examples in this chapter. A higher or lower threshold was used for a few images because the gray level range for those images was different.

The output of the program is a hierarchy of surface maps, one at each level of resolution. "Occluded" and "unknown" areas of the surface are marked. For each level, the output also includes a set of quadratic patches, as well as occluding and ridge contours.

5.1 Sphere Image

A 256x256 synthetic stereo pair of images of a sphere resting on a table is shown in Figure 25. To create this image, a real image of concrete was wrapped onto the sphere, and a real image of wood onto the table. The viewing directions were parallel. The disparity of the table ranges from 30 pixels at the rear to 65 pixels at the front, and the disparity of the closest point of the sphere is about 75 pixels. The contours and quadratic patches for the 256x256 level are shown in Figure 26. The patches are centered at the locations of the star-like objects. The contours around the sphere are all occluding contours, and the thick band along the left side of the sphere represents the area of the left image that is occluded, *i.e.*, not visible in the right image. Two views of the final disparity surface at the 256x256 level of resolution are shown in Figure 27. Areas of the surface which are "unknown" are assigned the lowest height for display purposes — no patches could be fit to these areas, nor were there known patches in the

vicinity to interpolate from. "Occluded" areas are displayed as a height slightly above "unknown". The surface appears to have a "hole" where depth values are unknown. The viewpoint for displaying the reconstructed surface is 25° above the horizontal plane, and ±25° from the axis running vertically through the image. This is the same for all the examples shown.

The ideal surface is shown in Figure 28. The difference between the ideal surface and the reconstructed surface is shown in Figure 29. Points which are different from the ideal surface by more than one pixel of disparity are shown in Figure 30. As in the baseball example, most of the large errors are due to a small misplacement of the occluding boundary.

5.2 Cube Image

A 256x256 synthetic stereo pair of images of a cube is shown in Figure 31. A random dot texture was mapped onto the faces of the cube. The left and right viewpoints were placed symmetrically on each side of one diagonal of the cube, and the viewing directions were parallel and at an angle of 30° to the top face of the cube. The background has a constant intensity value. The disparity of the farthest corner of the cube is about 30 pixels, and the disparity of the closest corner is about 45 pixels. The contours and quadratic patches for the 256x256 level are shown in Figure 32. The contours along the edges of the cube are all ridge contours. Two views of the final disparity surface at the 256x256 level of resolution are shown in Figure 33. The ideal surface is shown in Figure 34. The difference between the ideal surface and the reconstructed surface is shown in Figure 35. All points were within one pixel of the ideal surface. Points which had an absolute error of between 0.5 and 1 pixels of disparity are shown in Figure 36. Since there are no occluding contours ih this scene, there were no large errors.

5.3 Cone Image

A 512x512 synthetic stereo pair of images of a cone and a cube is shown in Figure 37. The objects are resting on a table and are in front of a wall. The cube partially occludes the cone. The textures used for this scene were taken from Brodatz [Brod56] and are: cork for the wall and table, brick for the cube, and reptile skin for the cone. The view directions were parallel. The disparity of the wall is about 50 pixels, the closest face of the cube is about 80 pixels, and the tip of the cone is about 60 pixels. The contours and quadratic patches for the 512x512 level are shown in Figure 38. There are ridge contours along the top edge of the cube and between the wall and the table. The other contours are all occluding contours. Two views of the final disparity surface at the 512x512 level of resolution are shown in Figure 39. The ideal surface is shown in Figure 40. The difference between the ideal surface and the reconstructed surface is shown in Figure 41. Points which had an absolute error greater than 1 pixel of disparity are shown in Figure 42. As before, most of the large errors are near the occluding boundaries.

Another way to display the resulting surface is to encode the height as an intensity value. Figure 43 shows the resulting surface and the ideal surface as intensity images. Finally, Figure 44 shows the status of the reconstructed surface. In this figure, black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

One can see a large unknown area on the top face of the cube, and Figure 38 shows that not many patches were fit in this area. This is because the brick texture in this area yields intensity edges which are predominantly horizontal. The algorithm does not attempt to match zero crossings which are near horizontal, because the disparity of horizontal zero crossings is subject to large error. Thus, there are not enough points in the area to fit patches, and so the surface is unknown there.

5.4 Ruts Image

A 512x512 real stereo pair of images of some ruts is shown in Figure 45. This scene has no occluding boundaries, but the tops of the ruts appear as sharp ridge boundaries. The images were taken with a 35 mm camera at a construction site just outside of our laboratory, and were digitized from the negatives. The camera had a 50 mm lens. The viewpoints were at a height of about 5 feet above the ground, looking obliquely downward, and their separation was about 1 foot. The view directions were not parallel, so the images were manually corrected by the method described earlier so that they were vertically registered. The disparity at the top of the image is about -7 pixels, and the disparity at the bottom of the image is about 60 pixels. The contours and quadratic patches for the 256x256 level are shown in Figure 46. Several ridge contours were found, along the tops of the ruts. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 47. Figures 48 and 49 show the corresponding results for the final 512x512 level. Figure 50 shows the resulting surface as an intensity image, and Figure 51 shows the status of the reconstructed surface. Although the ideal surface is not known for this example, the overall shape of the surface appears correct. One can see that few ridge contours were detected at the 512x512 level, although some were detected at the 256x256 level. This is probably due to the increased resolution at the 512x512 level — the tops of the ruts appear rounded at the higher level of resolution.

5.5 Rocks Image

A 512x512 real stereo pair of images of a mound of rocks and gravel is shown in Figure 52. This scene has a wide depth range, and has a significant occluding boundary at the far edge of the mound. The images were taken at the same construction site as the previous example, and were digitized from 35mm negatives. The viewpoints were at a height of about 5 feet above the ground, and the distance to the mound is on the order of 10 to 20 feet. The separation between the viewpoints was about 2 feet. The view directions were not parallel, so the images were manually corrected so that they were vertically registered. The disparity of the stereo pair ranges from about -65 pixels at the top of the image to about 85 pixels at the bottom of the image. The contours and quadratic patches for the 256x256 level are shown in Figure 53. An occluding contour was found at the far edge of the mound. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 54. Figures 55 and 56 show the corresponding results for the final 512x512 level. The overall shape of the surface appears correct.

5.6 Sandwich Image

A 512x512 real stereo pair of images of a peanut butter sandwich is shown in Figure 57. The images were taken with a 35 mm camera with a 50 mm lens, and were digitized from the negatives. The separation between the viewpoints was 7 cm, the human interocular distance. The view directions were parallel and the distance to the sandwich was approximately 1.5 feet. The disparity of the farthest point on the sandwich is about -10 pixels, and the disparity of the closest point is about 40 pixels. The contours and quadratic patches for the 256x256 level are shown in Figure 58. Several ridge contours were found. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 59. Figure 60 shows the resulting surface as an intensity image, and Figure 61 shows the status of the reconstructed surface. Figures 62 through 65 show the corresponding results for the final 512x512 level. The overall shape of the surface appears correct.

The background for this scene (everything except the sandwich) has no significant intensity texture, and generates very weak intensity edges. Most of the zero crossings detected for that region were weak and were eliminated by thresholding. With no threshold, isolated patches were occasionally fit in this region. These were incorrect, and appeared to be due to local groups of zero crossings happening to lie on a consistent surface, by chance. This is one of the few examples where thresholding the zero crossings made a difference.

5.7 Apple Image

A 512x512 real stereo pair of images of an Apple IIe mother board is shown in Figure 66. The images were taken with a TV camera with a 25 mm lens, and were digitized directly from the video signal. The separation between the viewpoints was about 6 inches and the distance to the board was approximately 1.5 feet. The view directions were not parallel, so the images were manually corrected so that they were vertically registered. The disparity of the stereo pair ranges from about -6 pixels in the upper left corner to about 6 pixels in the lower right corner. The contours and quadratic patches for the 256x256 level are shown in Figure 67. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 68. Figures 69 and 70 show the corresponding results for the final 512x512 level. Although the disparity range of this example is quite small, one can clearly make out the raised integrated circuit chips, and several ridge and occluding contours were found along their boundaries. Figure 71 shows the final 512x512 surface as an intensity image, and Figure 72 shows the status of the reconstructed surface.

The mother board does not appear planar in the disparity surface, as one would expect. However, it does not appear planar to us, either. This may be due to distortion in the wide angle (25 mm) lens used to take the pictures, or it may be due to the transformation used to vertically register the images, as the angle between the view directions for this example is larger than any of the other image pairs. The surface appears to be correct by manually checking at isolated points.

5.8 Books Image

A 512x512 real stereo pair of images of a stack of books is shown in Figure 73. The images were taken with a TV camera and were digitized directly from the video signal. The view directions were not parallel, so the images were manually corrected so that they were vertically registered. The disparity of the stereo pair ranges from about -30 pixels for the farthest point of the background to about 16 pixels for the closest point of the books. The contours and quadratic patches for the 256x256 level are shown in Figure 74. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 75. Figure 76 shows the 256x256 surface as an intensity image, and Figure 77 shows the status of the reconstructed surface. Figures 78 through 81 show the corresponding results for the final 512x512 level.

Although the overall surface appears correct, one can see that incorrect patches were fit in places, particularly in the upper left corner of the image, and along one edge of the stack of books. In carefully analyzing the results, it was found that the image is locally quite regular in those places, and this allows a consistent mis-matching of points. For example, there are several long straight edges on the books that are parallel and evenly spaced. Locally, the incorrect patches are the best fit to the data, but globally they are not. Some form of global constraint would be necessary to choose the correct patches in those places.

5.9 Fruit Image

A 512x512 real stereo pair of images of some fruit is shown in Figure 82. The images were taken with a 35 mm camera with a 50 mm lens, and were digitized from the negatives. The separation between the viewpoints was 7 cm, the human interocular distance. The view directions were approximately parallel, but it was necessary to manually correct the images so that they were vertically registered. In particular, the left side of the right image was stretched. The distance to the objects was on the order of 1 to 2 feet. The disparity of the stereo pair ranges from about -26 pixels at the top of the image to about 13 pixels at the bottom of the image. The contours and quadratic patches for the 256x256 level are shown in Figure 83. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 84. Figure 85 shows the 256x256 surface as an intensity image, and Figure 86

shows the status of the reconstructed surface. Figures 87 through 90 show the corresponding results for the final 512x512 level.

One can see in the 512x512 level results that there are large areas where surface patches could not be fit, particularly in the area of the cantaloupe. In analyzing this image, it was found that the images are not vertically registered there, but are misaligned by about 4 pixels. This is enough so that matches could not be found for the zero crossings in that region. Since the misalignment is relatively less in the 256x256 level (2 pixels), enough matches were found so that patches could be fit for that level. The misalignment may be due to the method used to vertically register the images, or it may be due to non-ideal camera optics. The reconstructed surfaces appear correct, in the 256x256 level and in the known areas of the 512x512 level. The various objects can be readily identified in the height maps.

5.10 Pentagon Image

A 512x512 real stereo pair of images of an aerial view of the Pentagon Building is shown in Figure 91. The images were obtained from Professor Takeo Kanade of the Carnegie-Mellon University Computer Science Department. The images are vertically registered. The disparity of the backgournd is about -5 pixels and the disparity of the top of the building is about 5 pixels. The contours and quadratic patches for the 512x512 level are shown in Figure 92. One can see that there are large areas which have no surface patches; for example, the area in the northernmost wing of the building. The reason for this is that there are many horizontal edges in this part of the image, and the algorithm cannot reliably match horizontal edges. Two views of the disparity surface at the final 512x512 level of resolution are shown in Figure 93. Figure 94 shows the 512x512 surface as an intensity image, and Figure 95 shows the status of the reconstructed surface. In the "known" areas, the surface appears to be correct.

5.11 Renault Image

A 512x512 real stereo pair of images of a Renault auto part is shown in Figure 96. The images were obtained from Professor Gerard Medioni of the USC Intelligent Systems Group, and were transformed so that they were vertically registered. Although the auto part is the main object in this scene, the table on which it is resting is slightly dirty and has enough texture so that fusion of that area is possible. The disparity of the auto part ranges from about 6 pixels at the leftmost tip to about 30 pixels at the bottom. The contours and quadratic patches for the 256x256 level are shown in Figure 97. Two views of the disparity surface at the 256x256 level of resolution are shown in Figure 98. Figure 99 shows the 256x256 surface as an intensity image, and Figure 100 shows the status of the reconstructed surface. Figures 101 through 104 show the corresponding results for the final 512x512 level. The surface appears to be correct. Even most of the surface of the table was found.

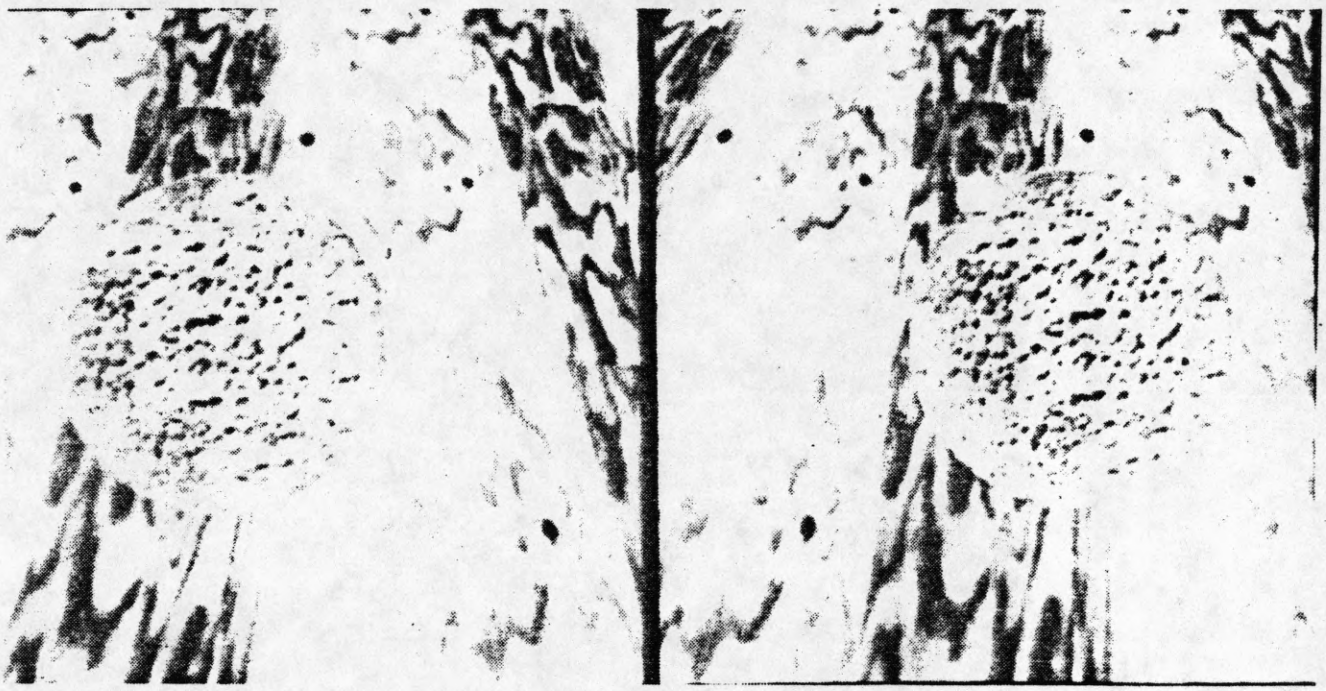

Figure 25. A 256x256 synthetic image of a sphere on a table. The disparity of the table ranges from 30 pixels at the rear to 65 pixels at the front, and the disparity of the closest point of the sphere is about 75 pixels.

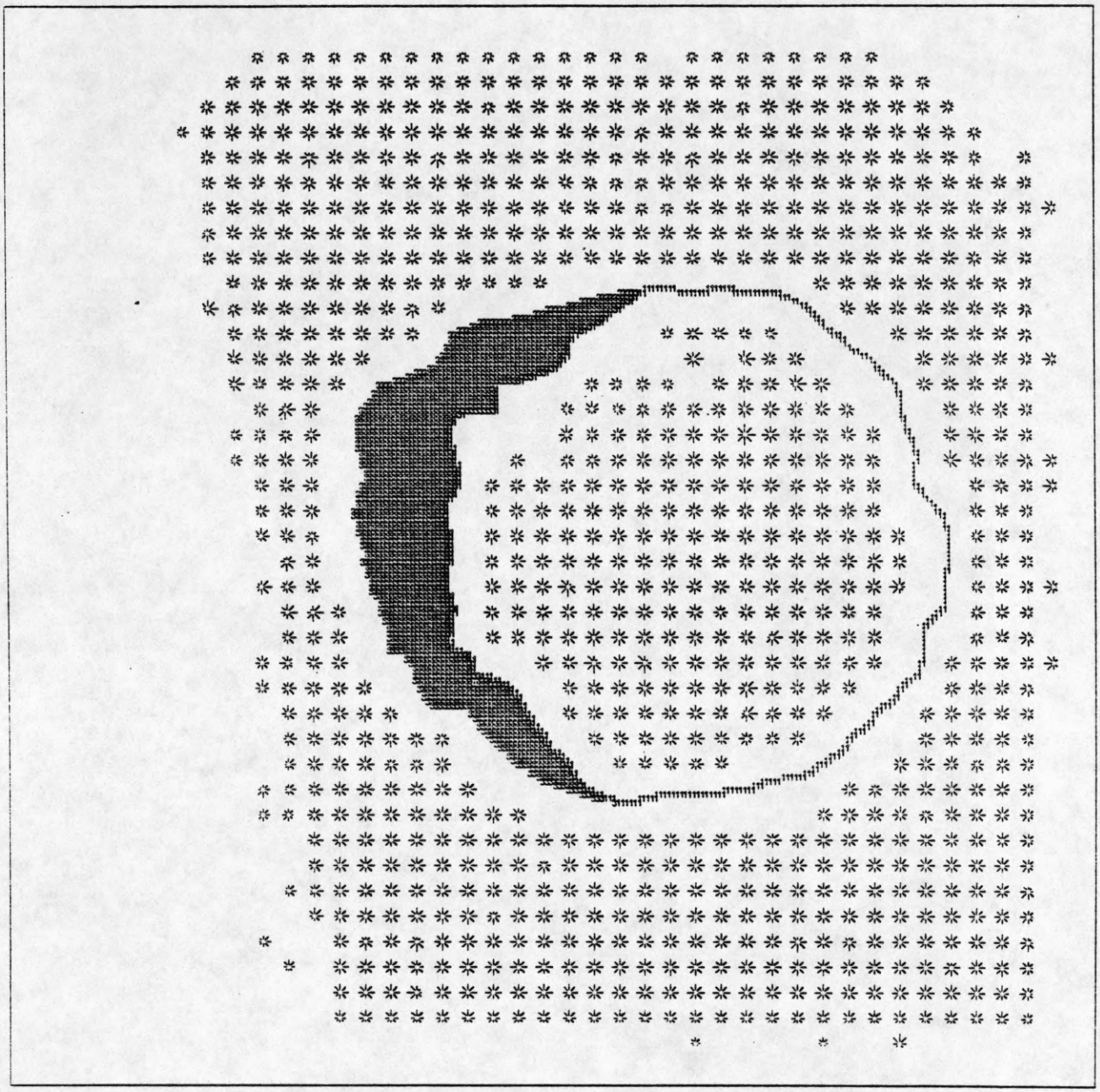

Figure 26. Quadratic patches and contours found for the sphere image, at the 256x256 level of resolution. The contours are all occluding boundaries and the thick band on the left of the sphere shows an occluded region.

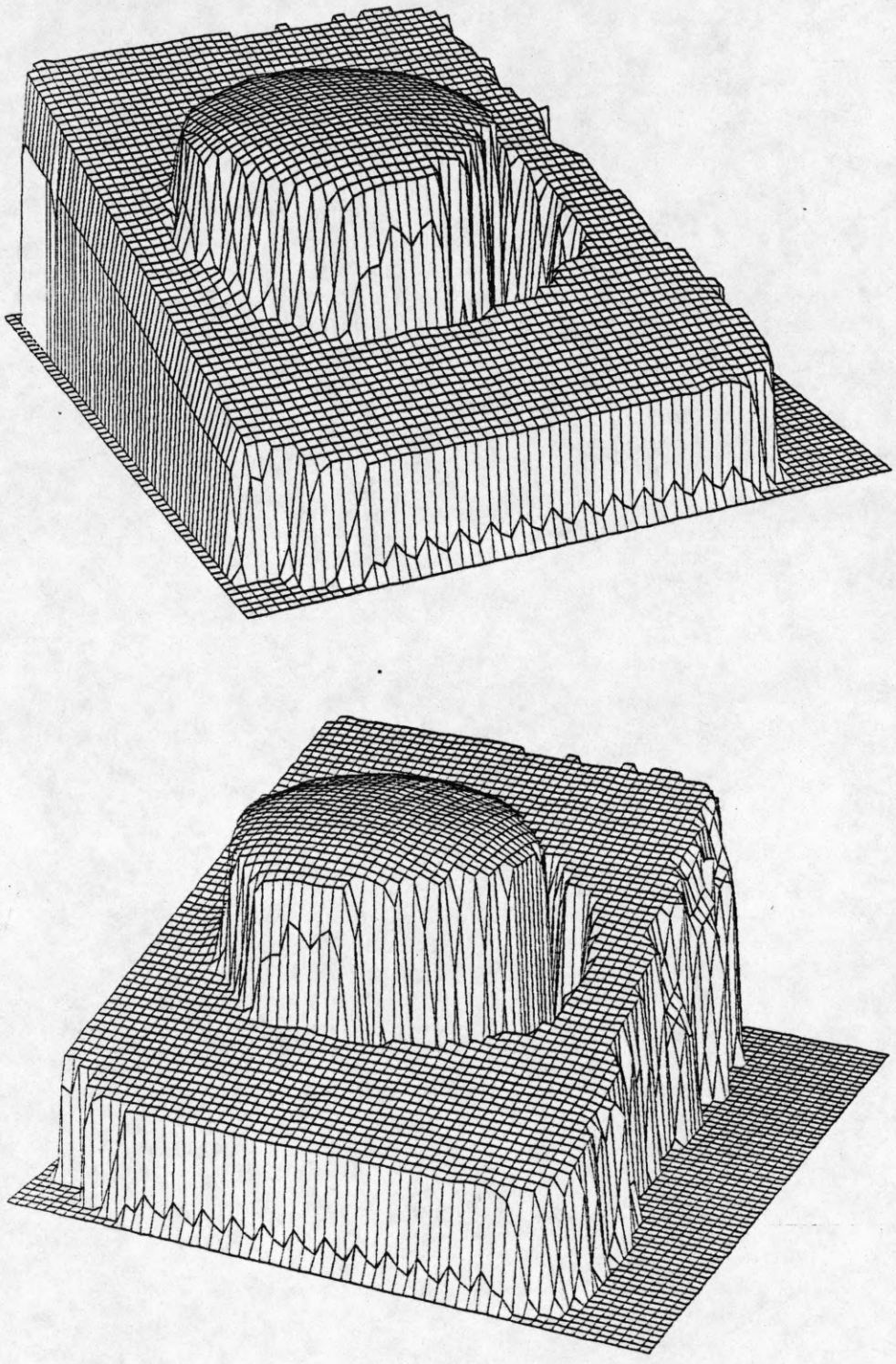

Figure 27. Reconstructed disparity surface for the sphere image, at the 256x256 level of resolution. Disparity ranges from 5 to 76 pixels.

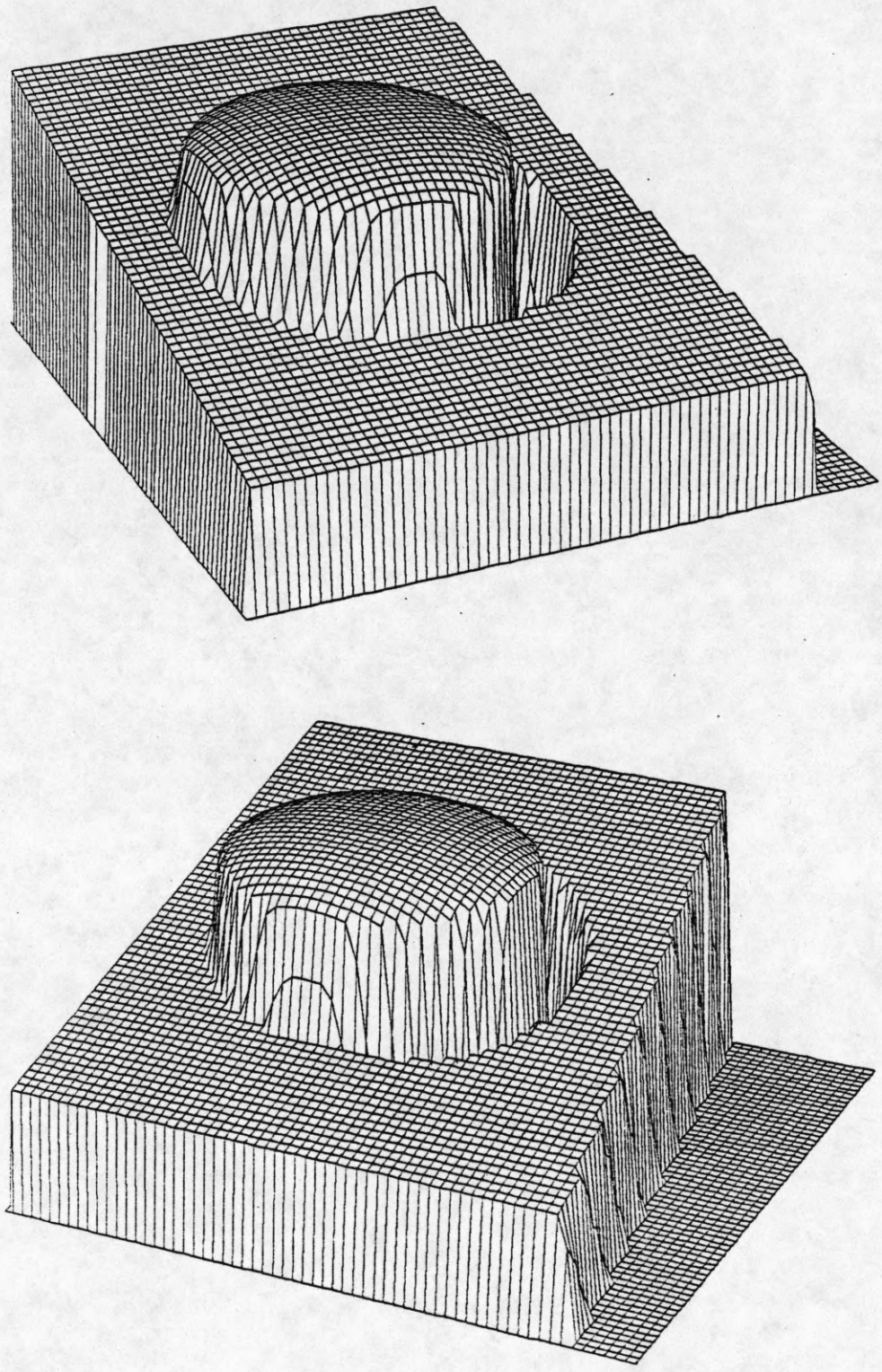

Figure 28. Ideal disparity surface for the sphere image. Disparity ranges from 0 to 76 pixels.

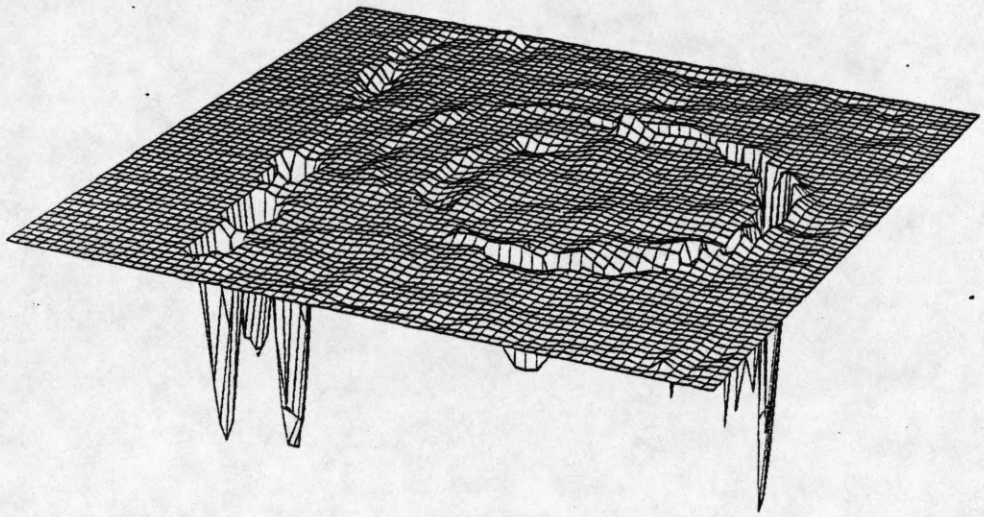

Figure 29. Difference between ideal and calculated disparity surfaces, for the 256x256 level. Disparity ranges from -21 to 2 pixels.

Figure 30. Points which have a disparity error magnitude greater than one pixel, for the 256x256 level of resolution.

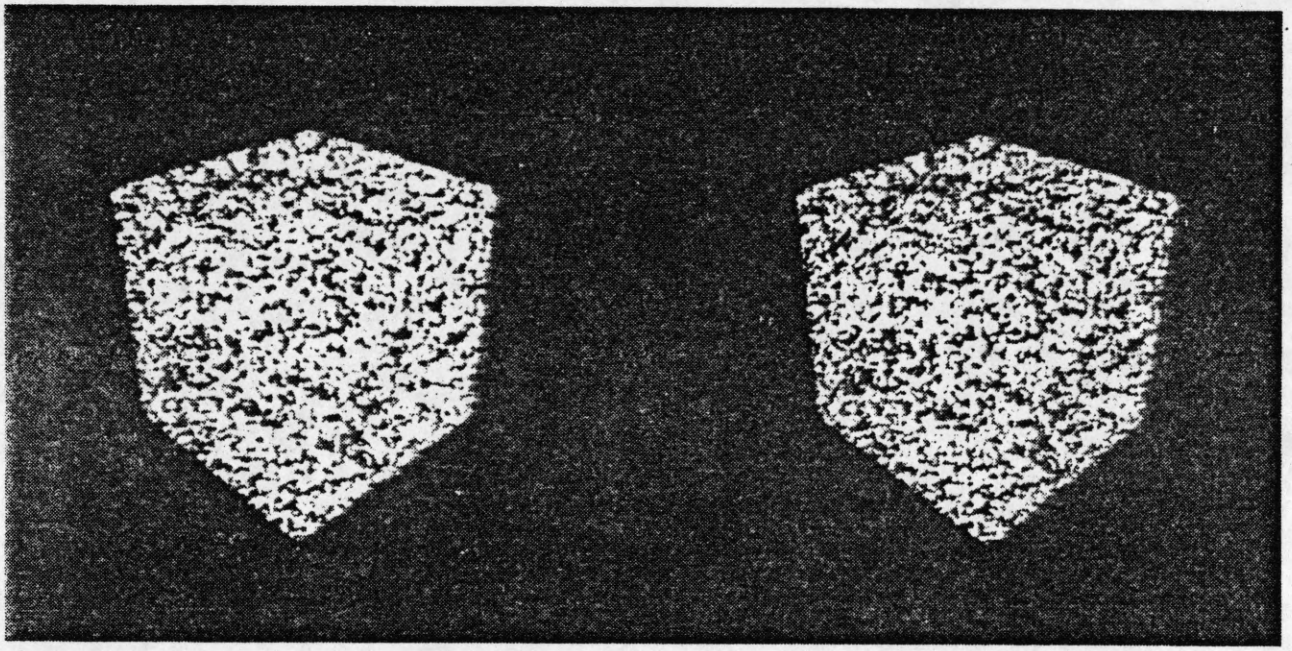

Figure 31. A 256x256 synthetic image of a cube. The disparity of the farthest corner of the cube is about 30 pixels, and the disparity of the closest corner is about 45 pixels.

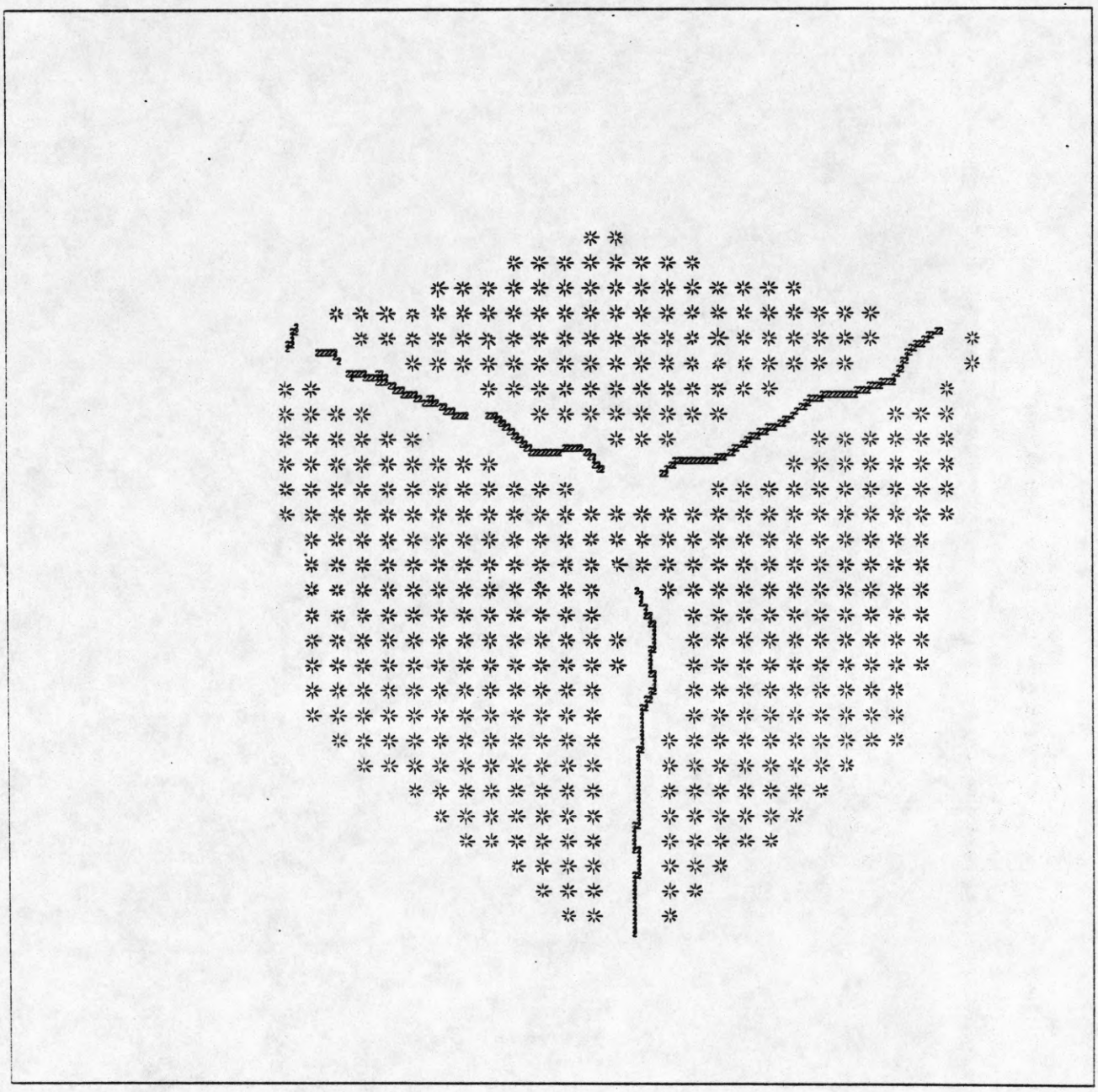

Figure 32. Quadratic patches and contours found for the cube image, at the 256x256 level of resolution. The contours are all ridge boundaries.

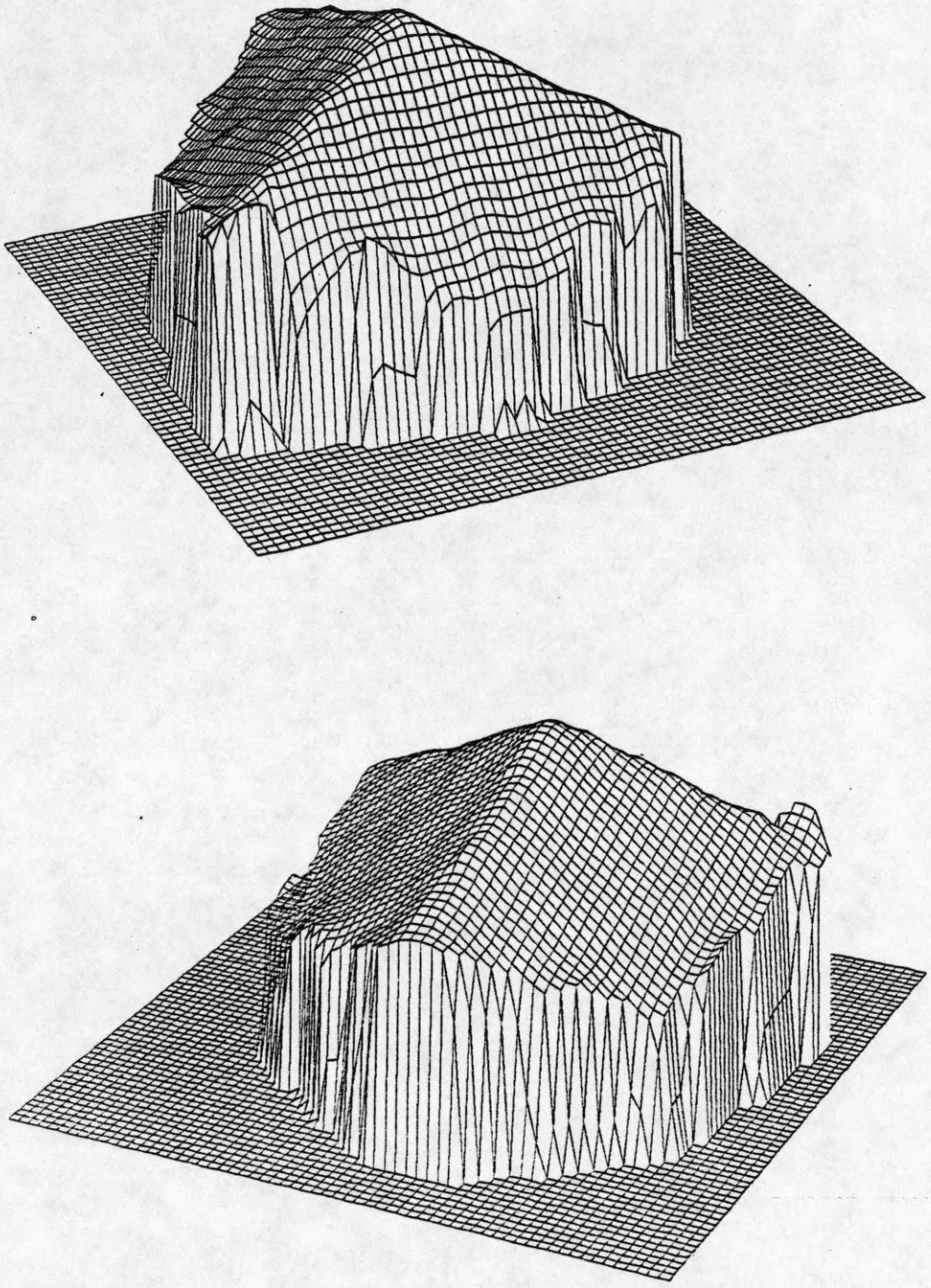

Figure 33. Reconstructed disparity surface for the cube image, at the 256x256 level of resolution. Disparity ranges from 17 to 45 pixels.

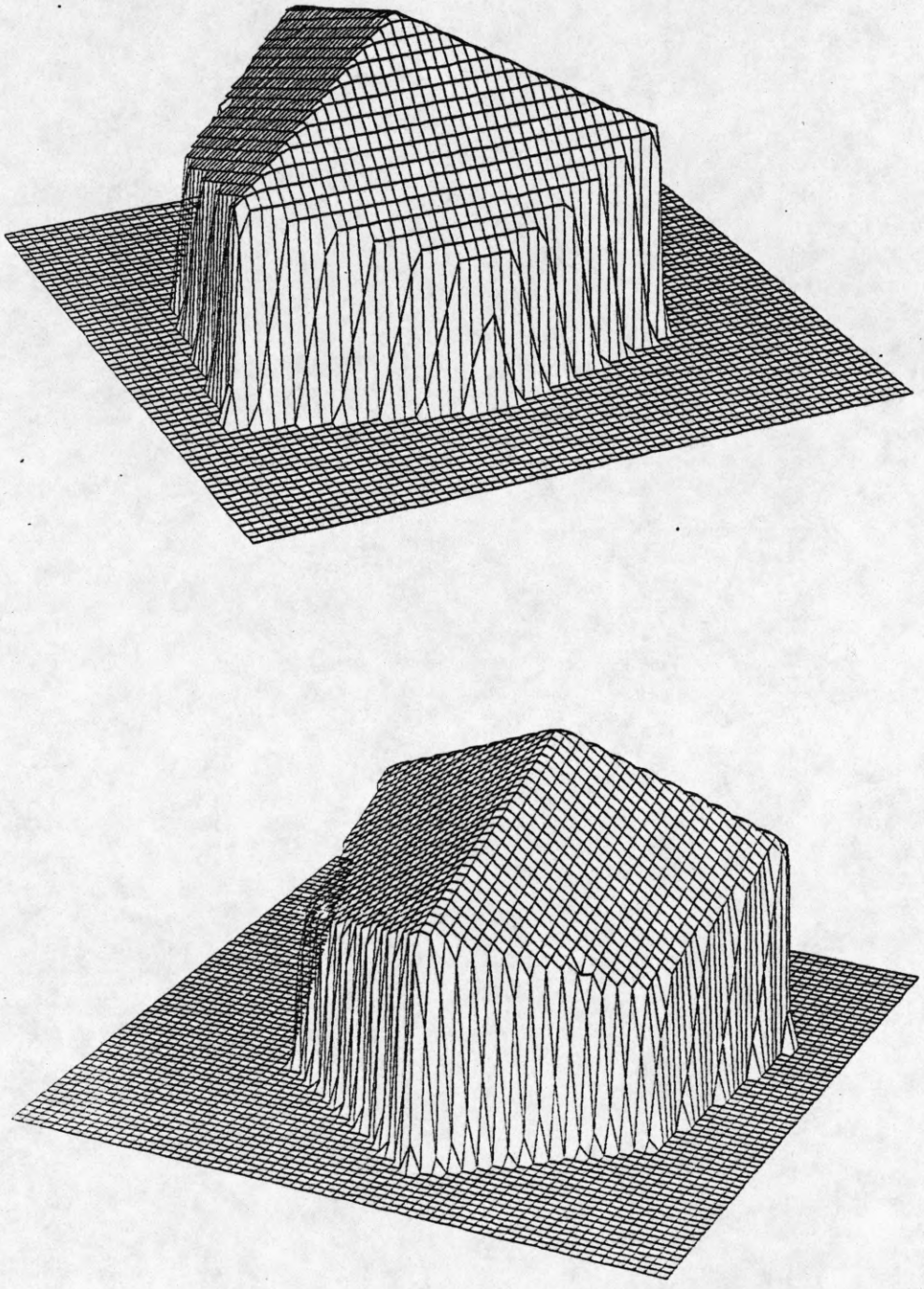

Figure 34. Ideal disparity surface for the cube image. Disparity ranges from 17 to 45 pixels.

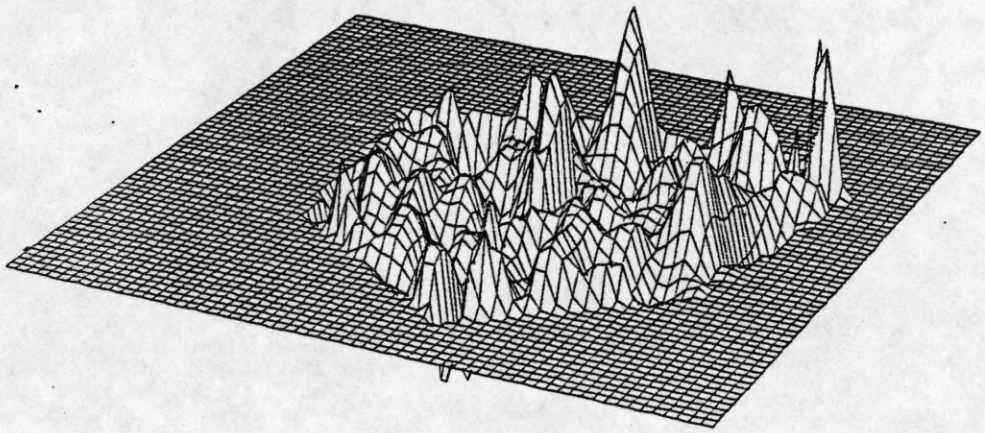

Figure 35. Difference between ideal and calculated disparity surfaces, for the 256x256 level. Disparity ranges from -0.8 to 0.6 pixels.

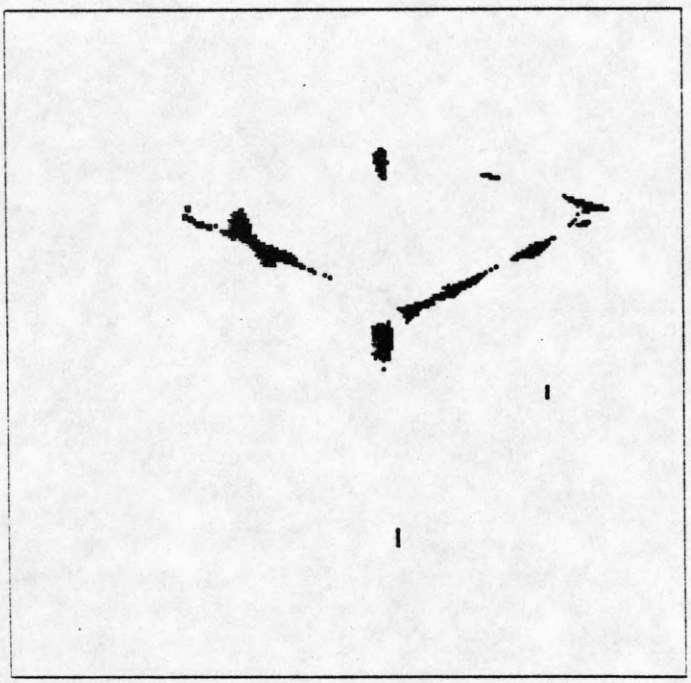

Figure 36. Points which have a disparity error magnitude greater than 0.5 pixel, for the 256x256 level of resolution.

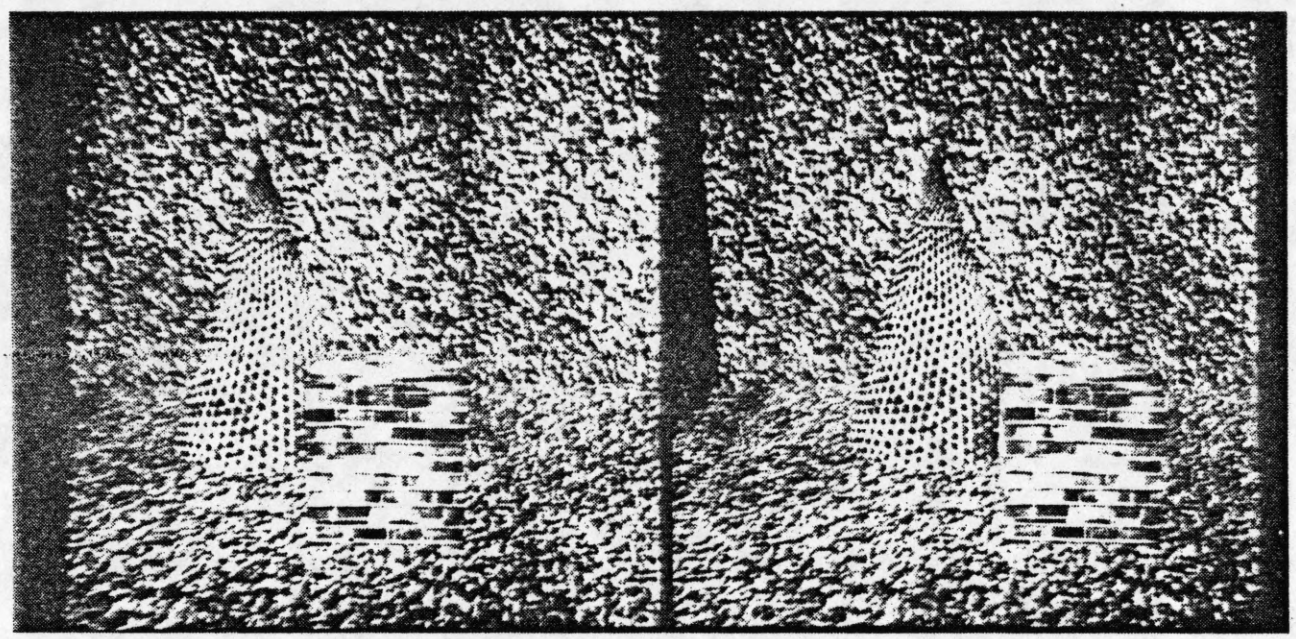

Figure 37. A 512x512 synthetic image of a cone and a cube on a table, against a background of a wall. The disparity of the wall is about 50 pixels, the closest face of the cube is about 80 pixels, and the tip of the cone is about 60 pixels.

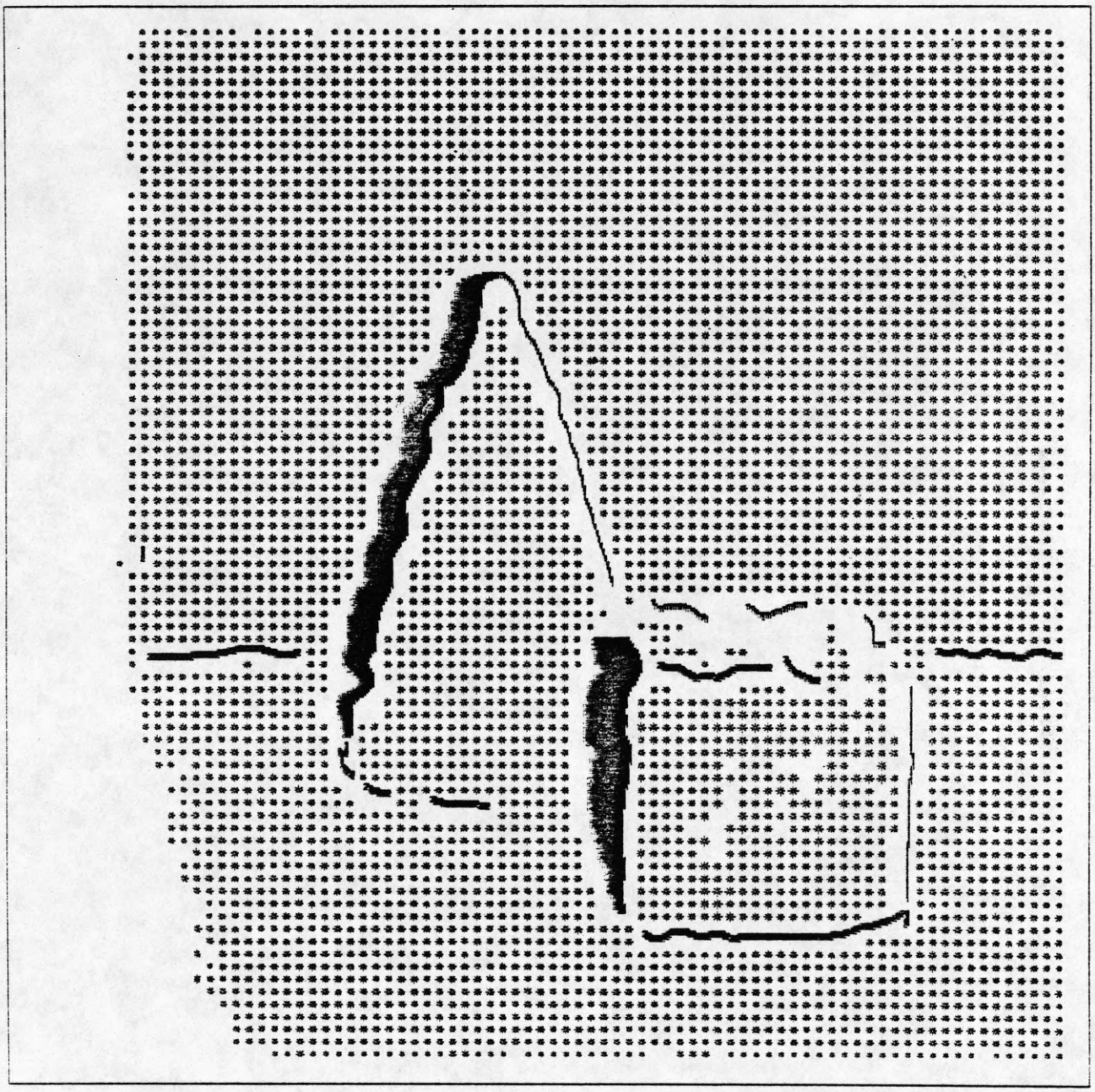

Figure 38. Quadratic patches and contours found for the cone image, at the 512x512 level of resolution.

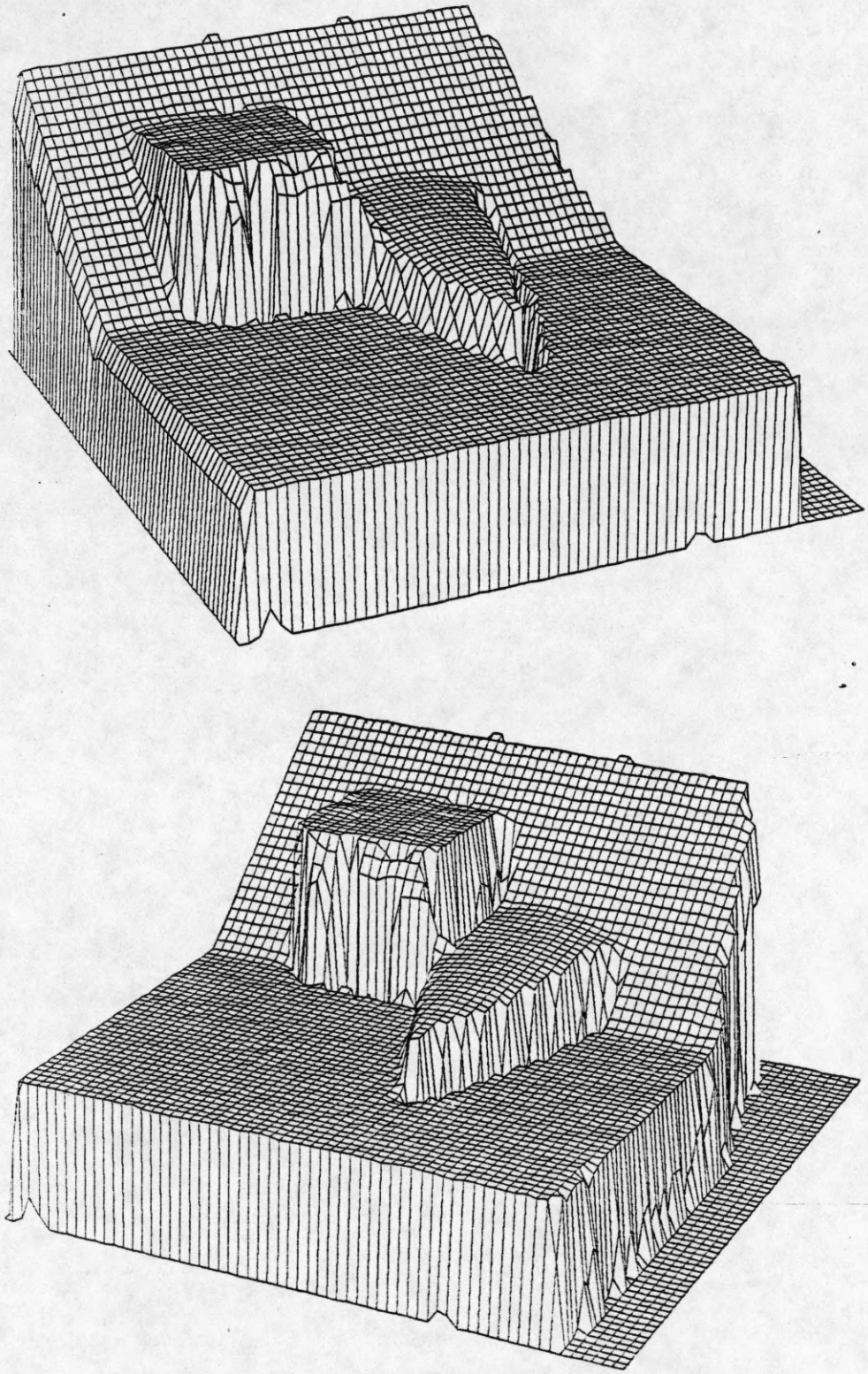

Figure 39. Reconstructed disparity surface for the cone image, at the 512x512 level of resolution. Disparity ranges from 15 to 93 pixels.

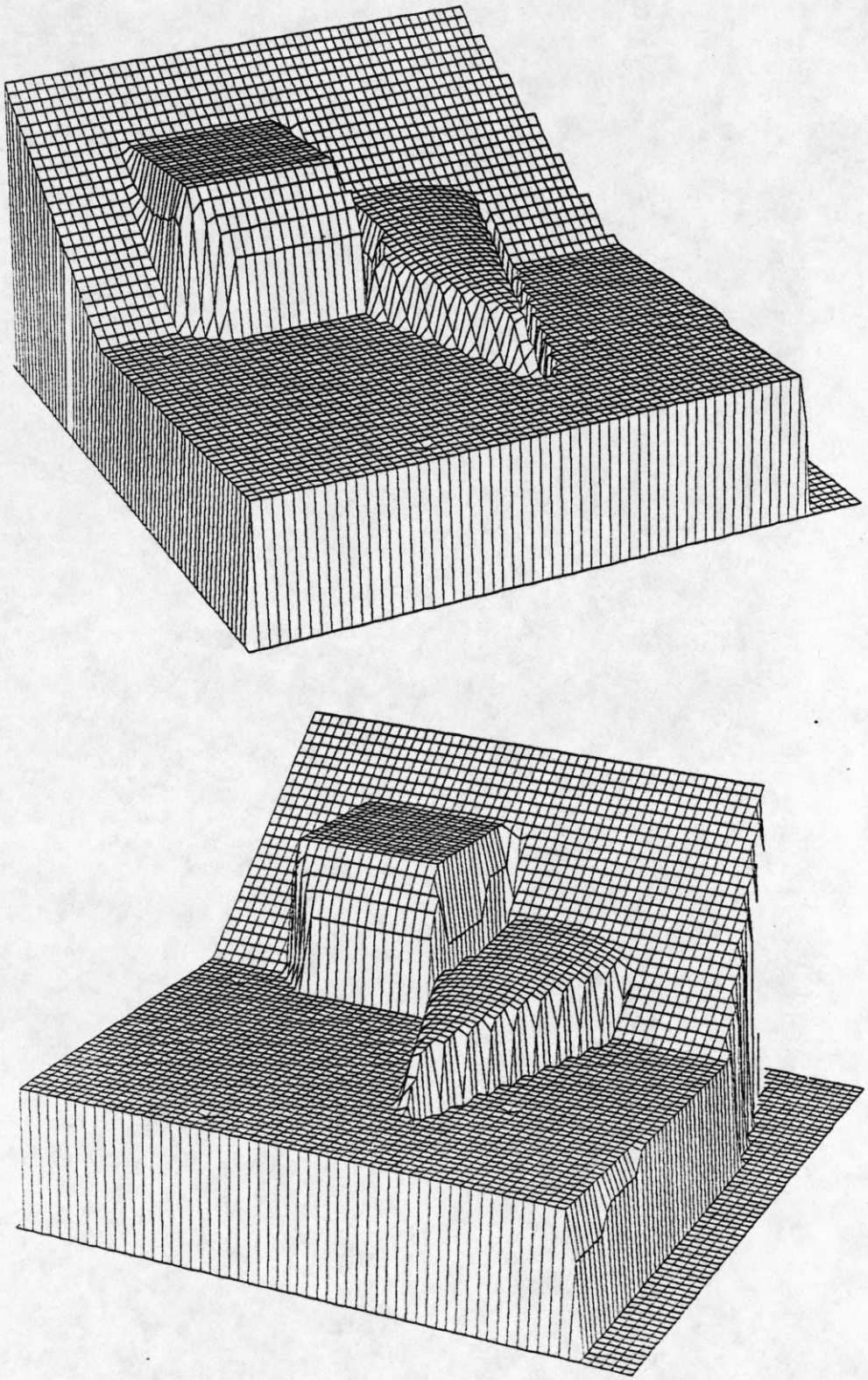

Figure 40. Ideal disparity surface for the cone image. Disparity ranges from 15 to 94 pixels.

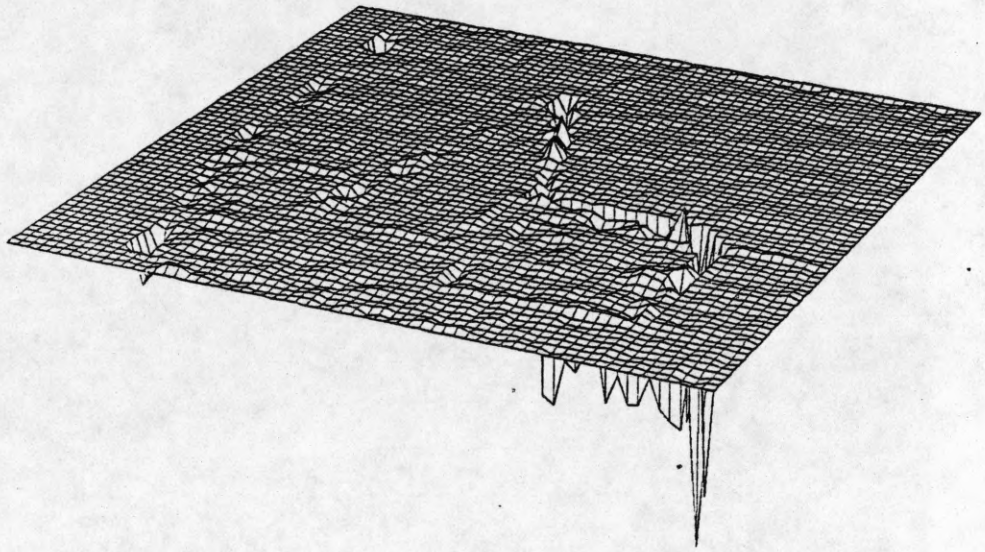

Figure 41. Difference between ideal and calculated disparity surfaces, for the 512x512 level. Disparity ranges from -32 to 3 pixels.

Figure 42. Points which have a disparity error magnitude greater than 1 pixel, for the 512x512 level of resolution.

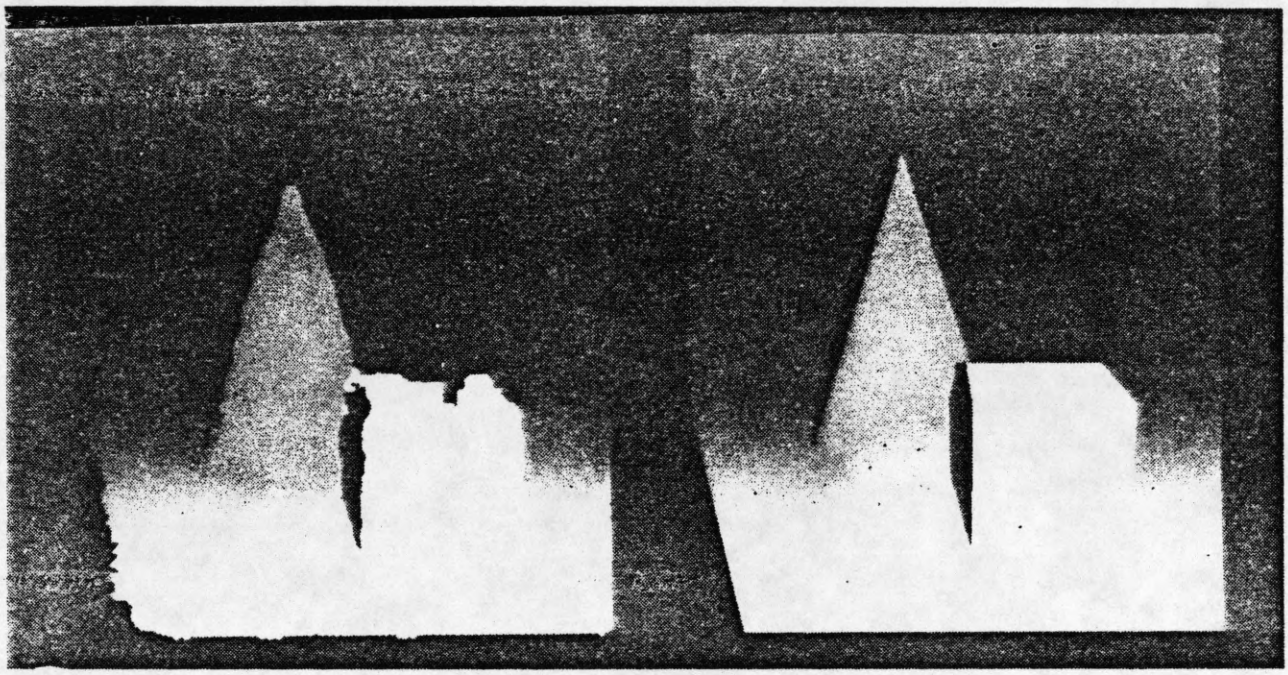

Figure 43. Ideal and calculated disparity surfaces shown as intensity images.

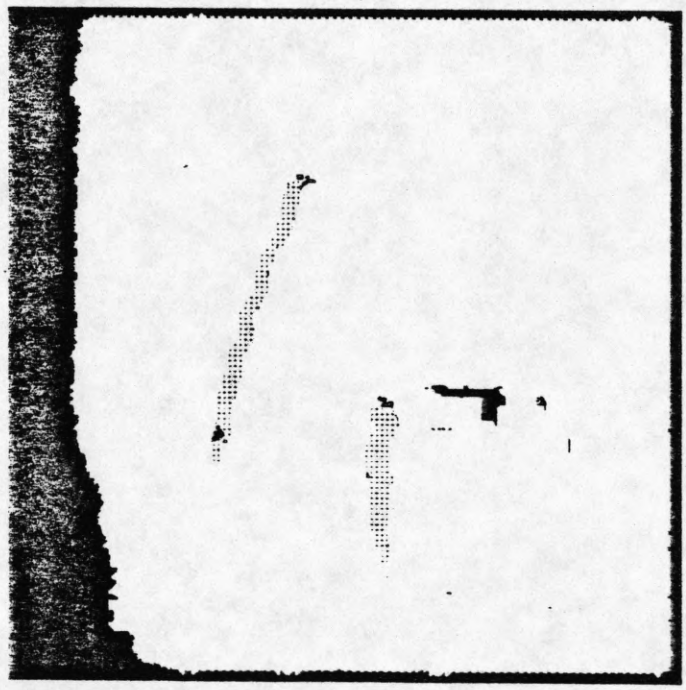

Figure 44. Status of the reconstructed surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

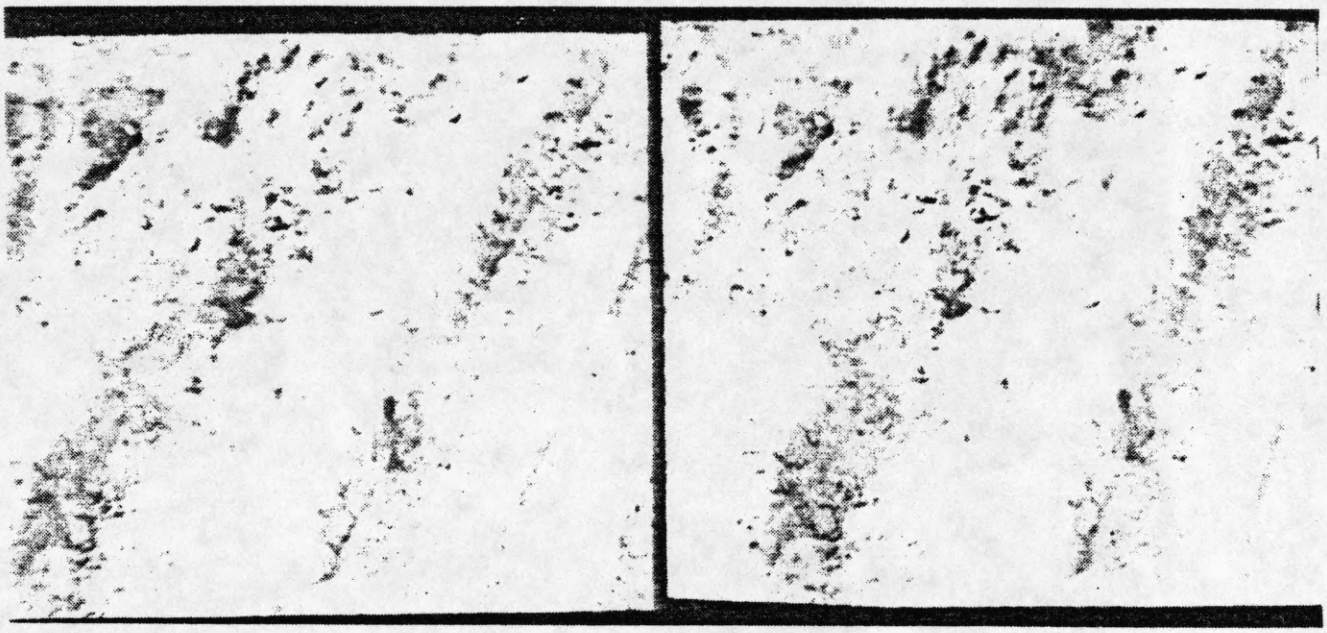

Figure 45. A 512x512 real stereo pair of images of some ruts. This scene has no occluding boundaries, but the tops of the ruts appear as sharp ridge boundaries. The disparity at the top of the image is about -7 pixels, and the disparity at the bottom of the image is about 60 pixels..

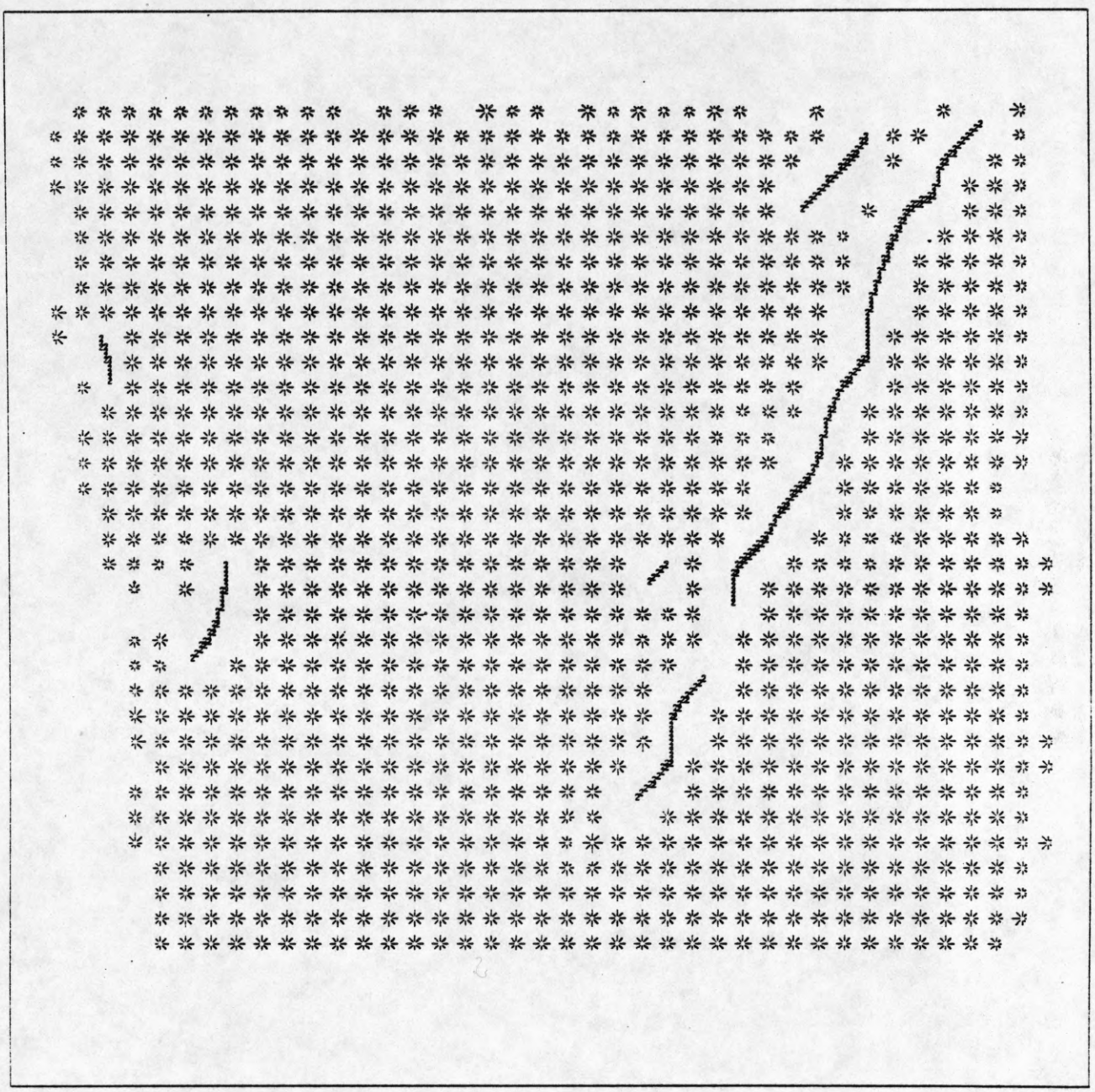

Figure 46. Quadratic patches and contours found for the ruts image, at the 256x256 level of resolution.

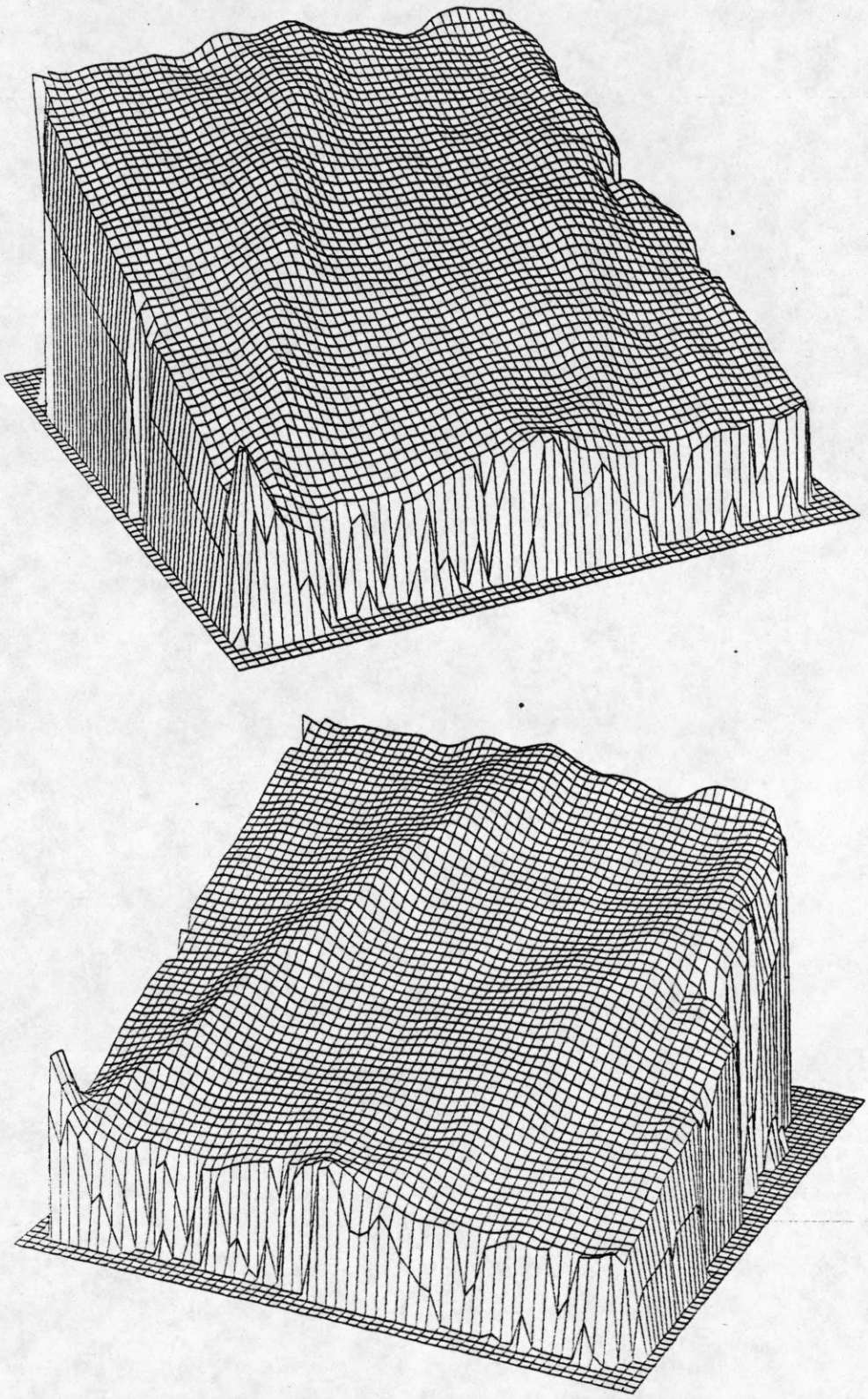

Figure 47. Reconstructed disparity surface for the ruts image, at the 256x256 level of resolution. Disparity ranges from -15 to 30 pixels.

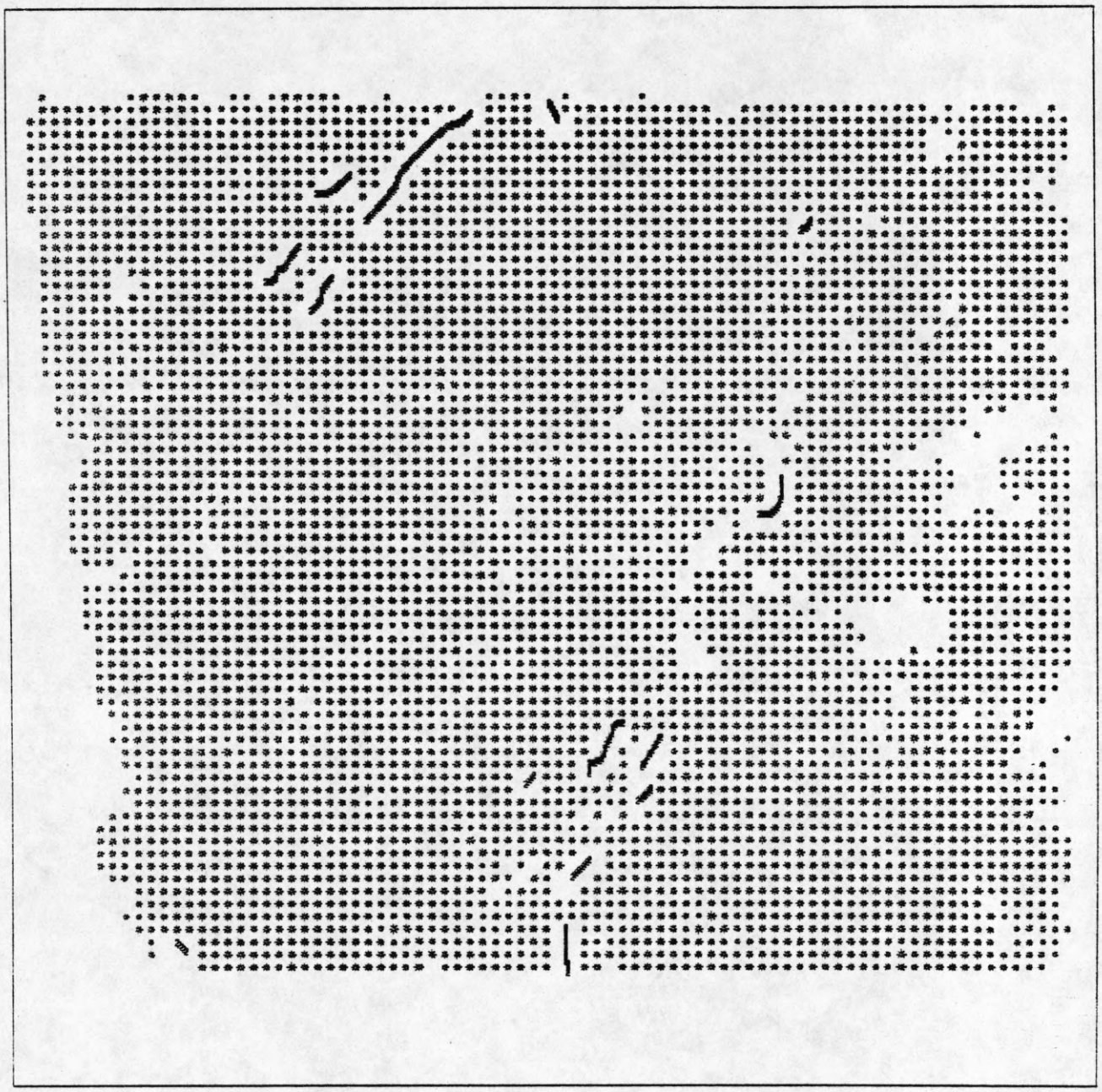

Figure 48. Quadratic patches and contours found for the ruts image, at the 512x512 level of resolution.

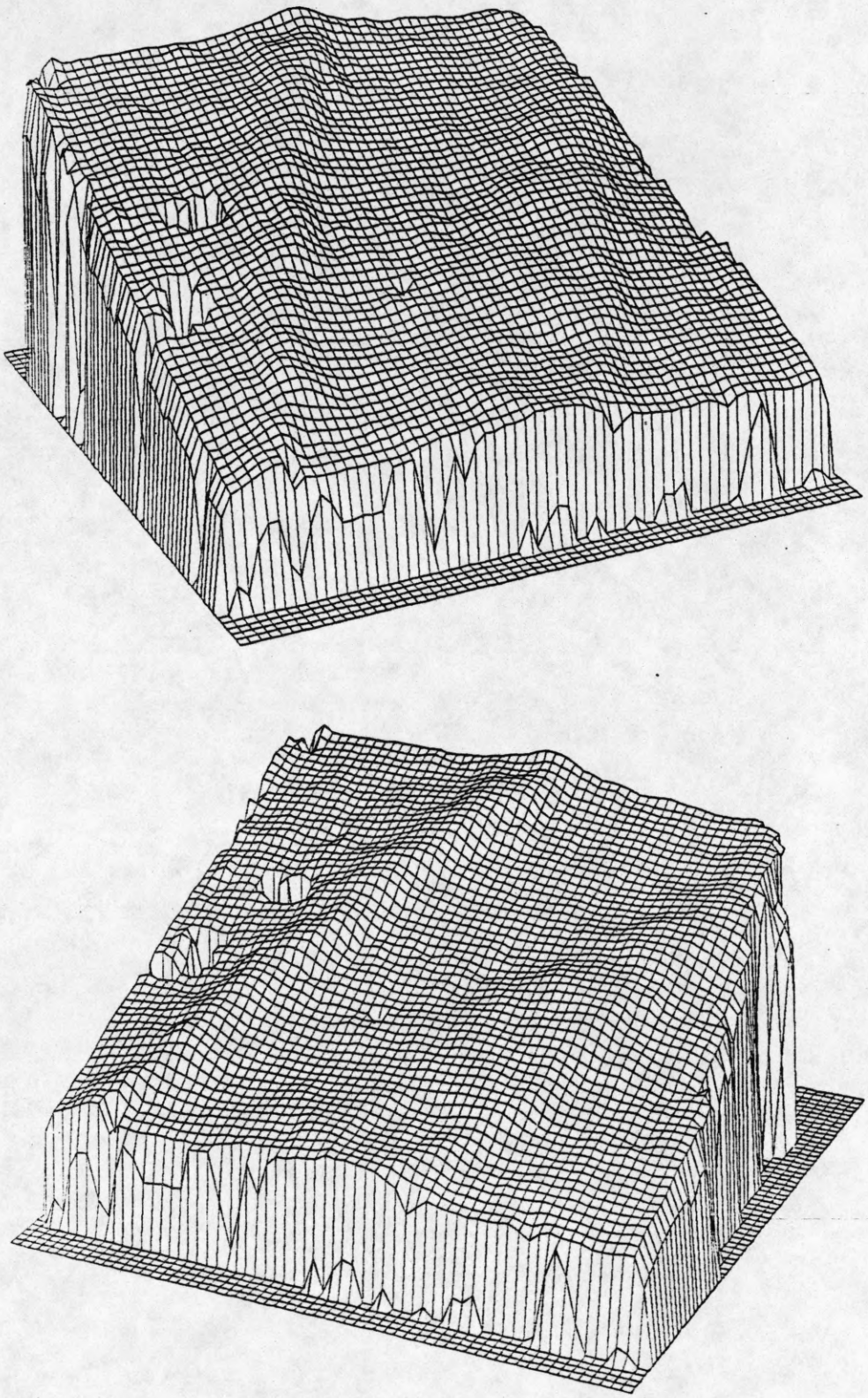

Figure 49. Reconstructed disparity surface for the ruts image, at the 512x512 level of resolution. Disparity ranges from -35 to 60 pixels.

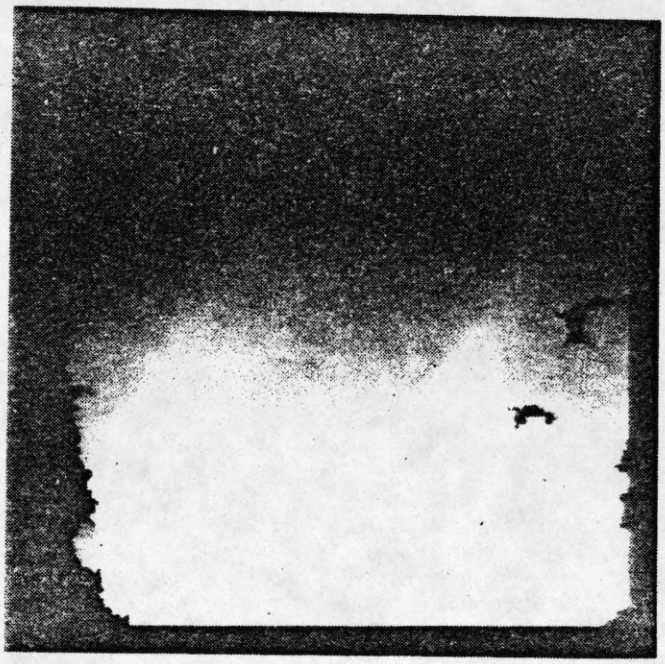

Figure 50. The 512x512 disparity surface shown as an intensity image.

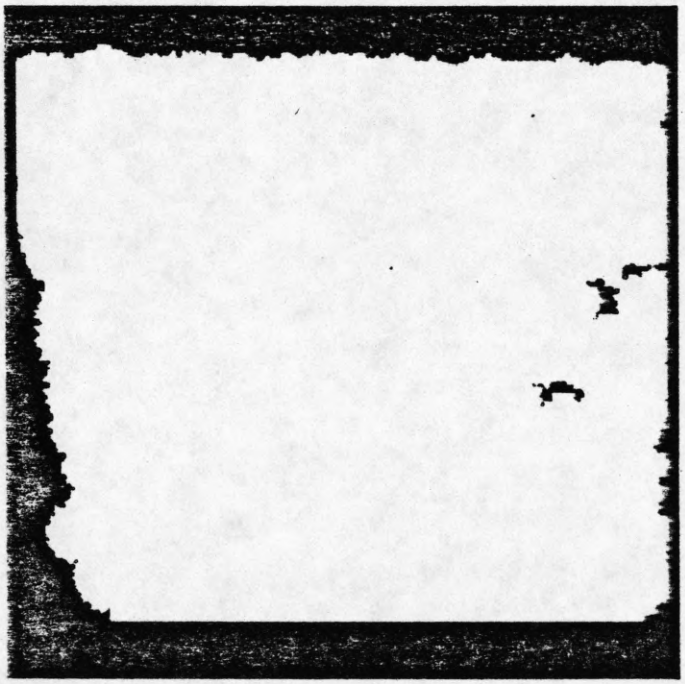

Figure 51. Status of the reconstructed surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

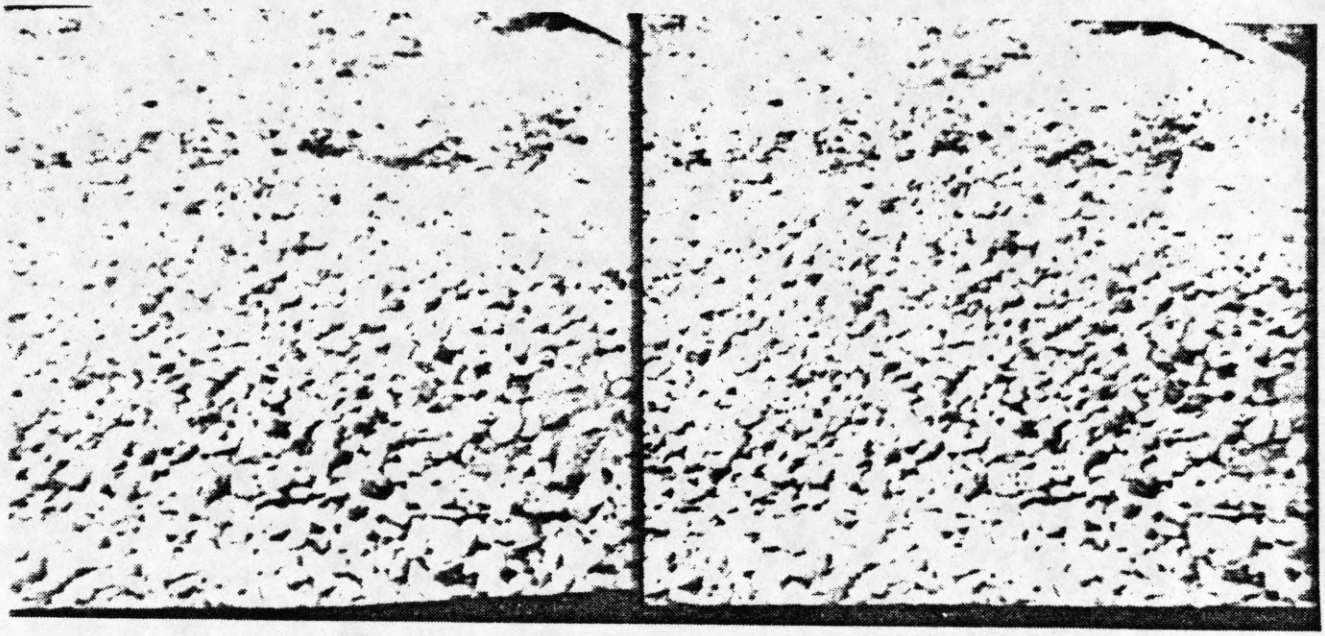

Figure 52. A 512x512 real stereo pair of images of a mound of rocks and gravel. This scene has a significant occluding boundary at the far edge of the mound. The disparity ranges from about -65 pixels at the top of the image to about 85 pixels at the bottom of the image..

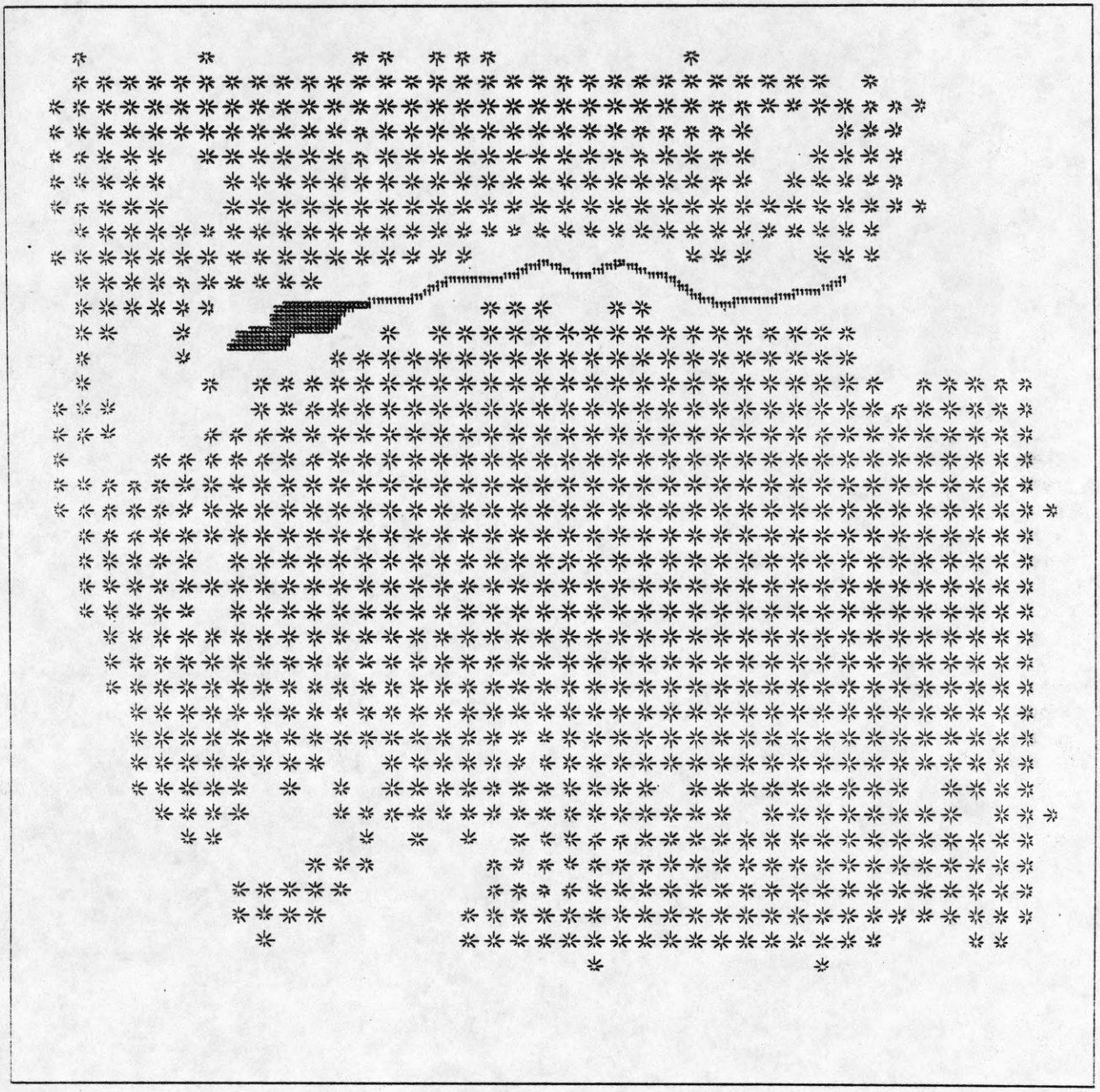

Figure 53. Quadratic patches and contours found for the rocks image, at the 256x256 level of resolution.

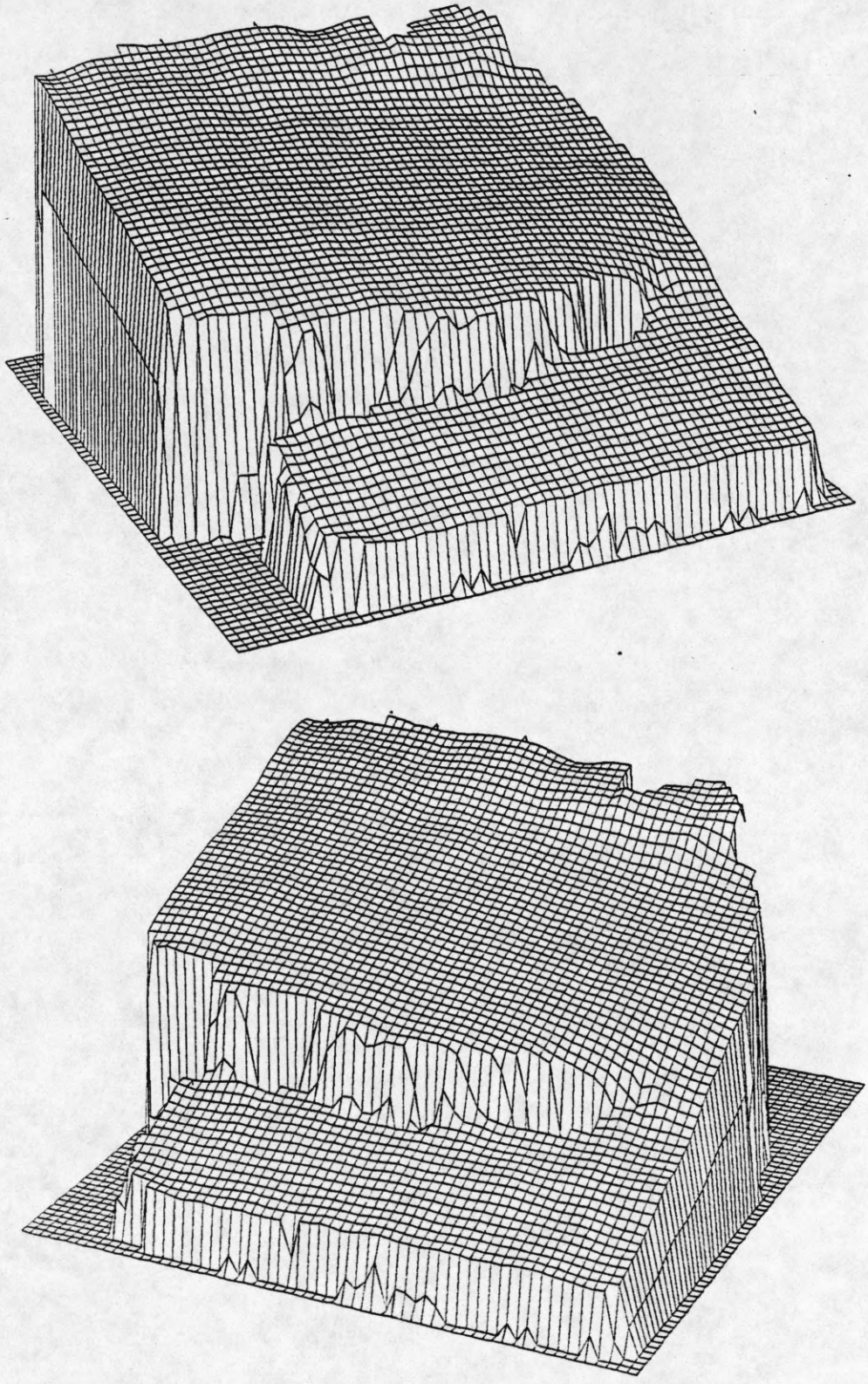

Figure 54. Reconstructed disparity surface for the rocks image, at the 256x256 level of resolution. Disparity ranges from -50 to 45 pixels.

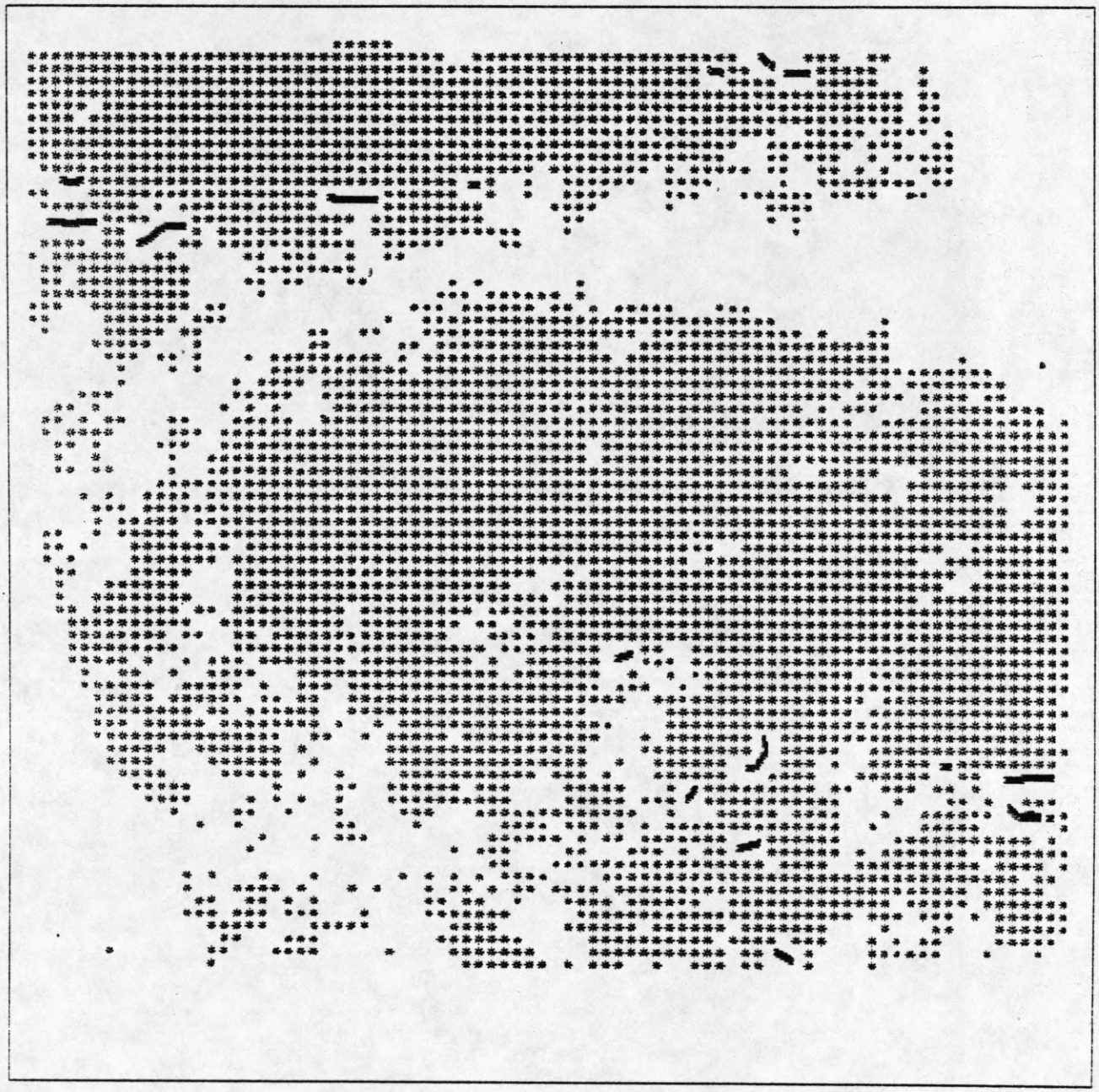

Figure 55. Quadratic patches and contours found for the rocks image, at the 512x512 level of resolution.

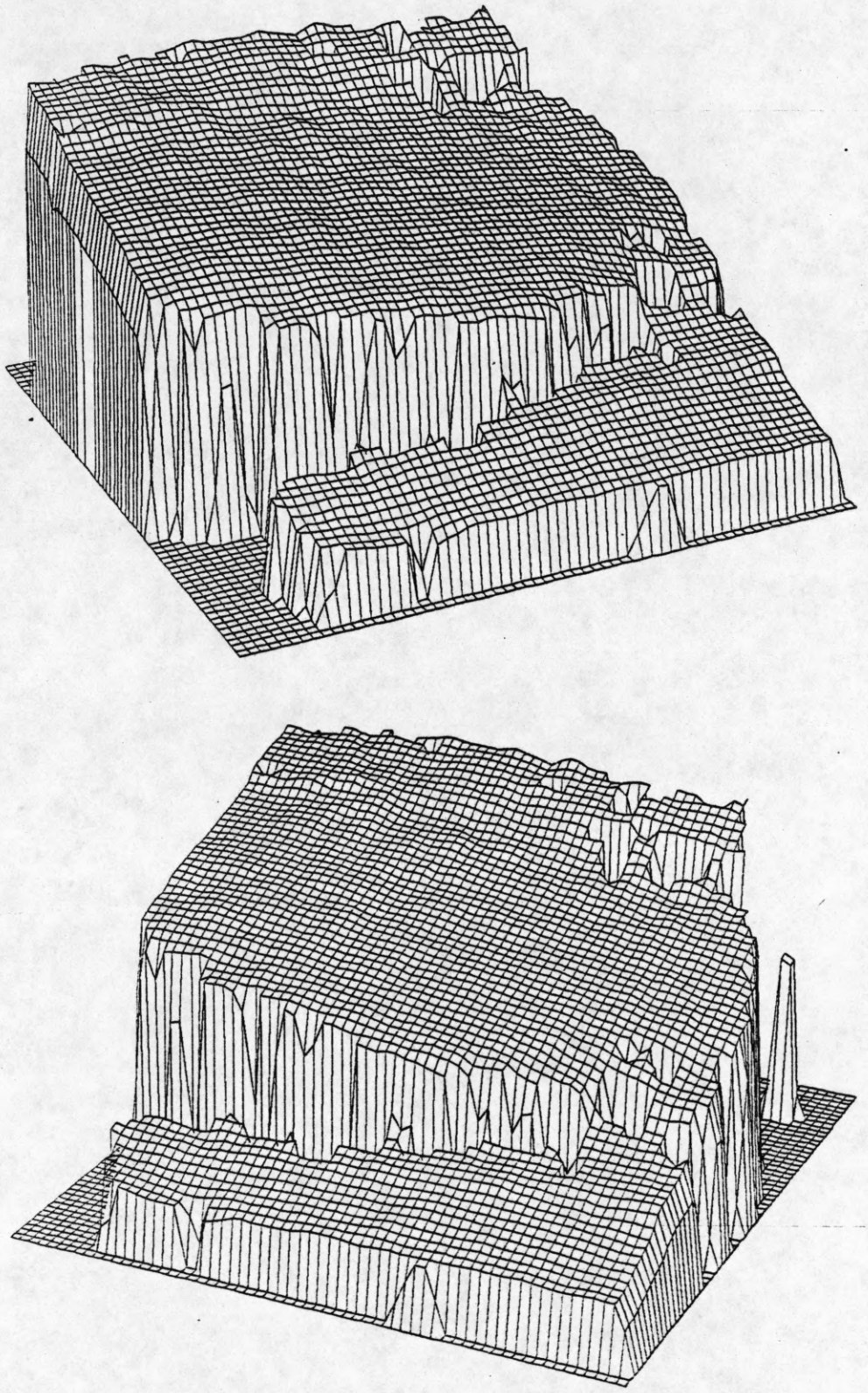

Figure 56. Reconstructed disparity surface for the rocks image, at the 512x512 level of resolution. Disparity ranges from -100 to 89 pixels.

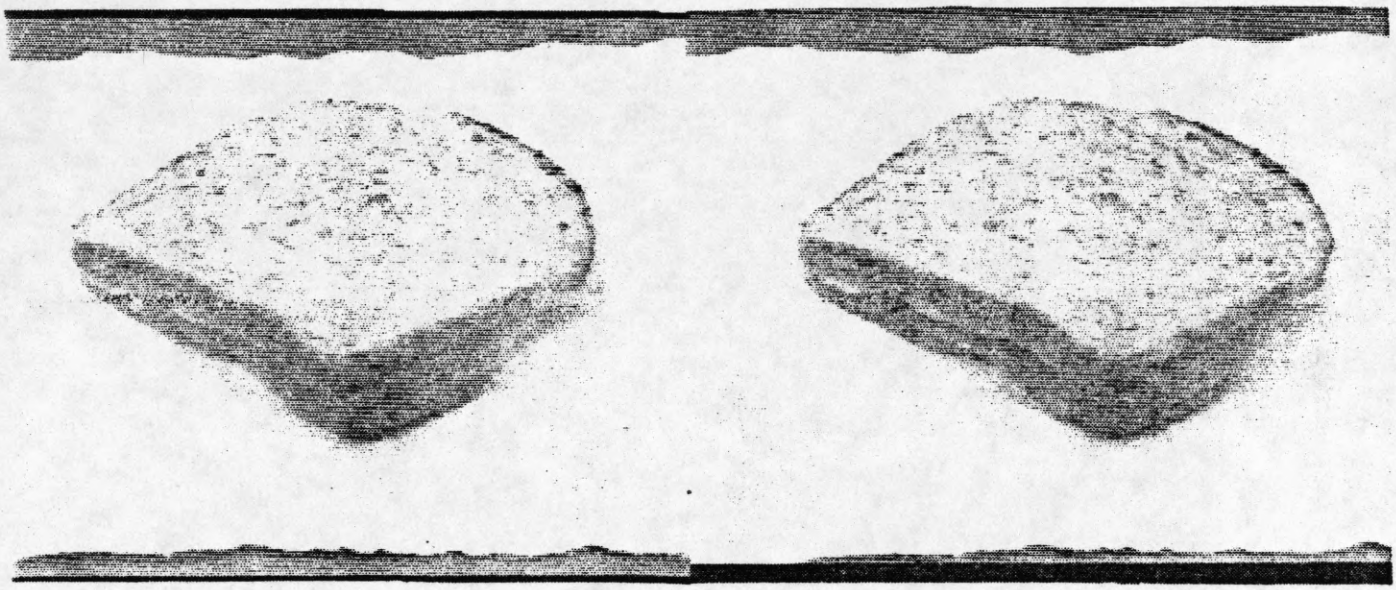

Figure 57. A 512x512 real stereo pair of images of a peanut butter sandwich. The disparity of the farthest point on the sandwich is about -10 pixels, and the disparity of the closest point is about 40 pixels.

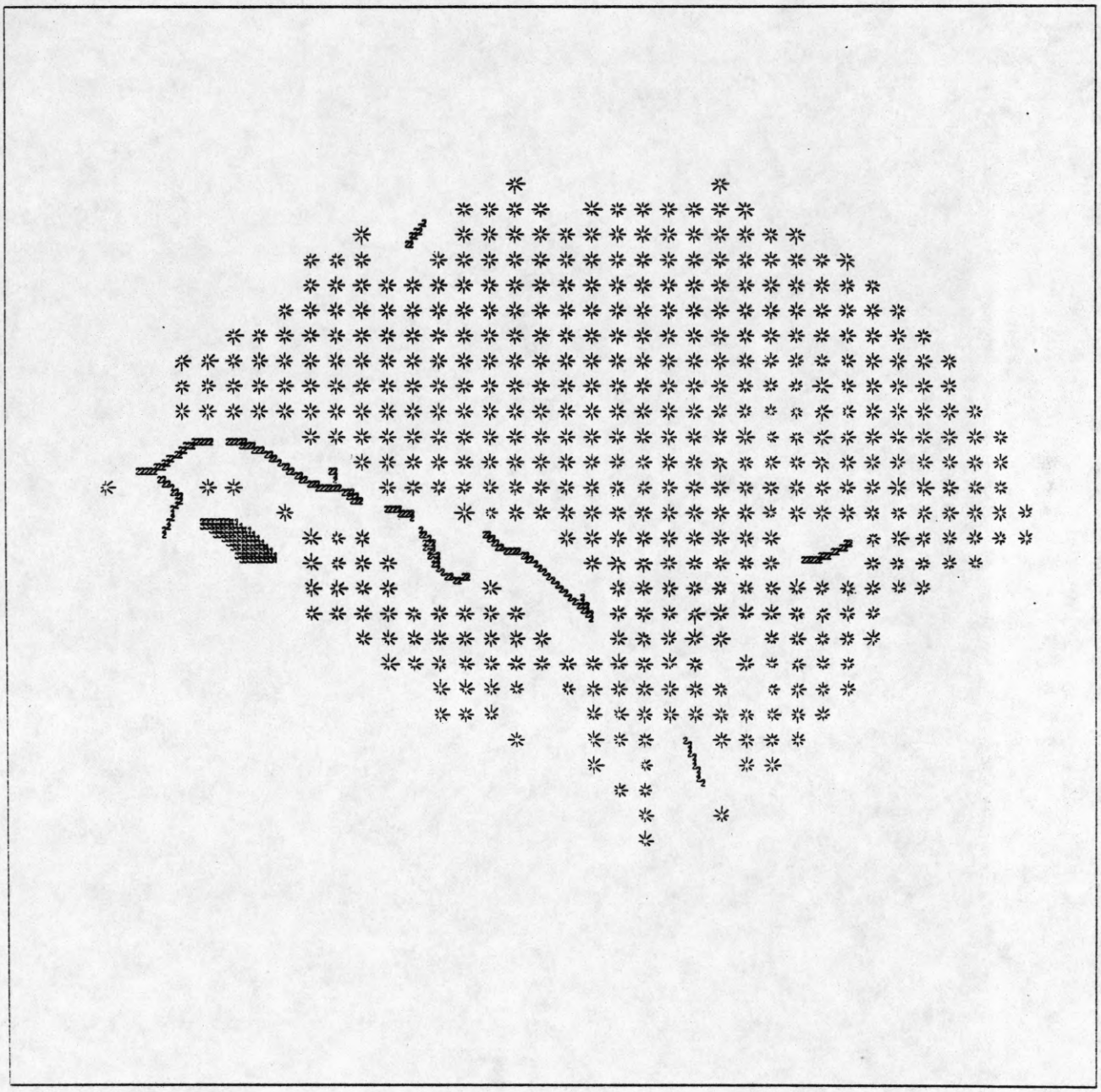

Figure 58. Quadratic patches and contours found for the sandwich image, at the 256x256 level of resolution.

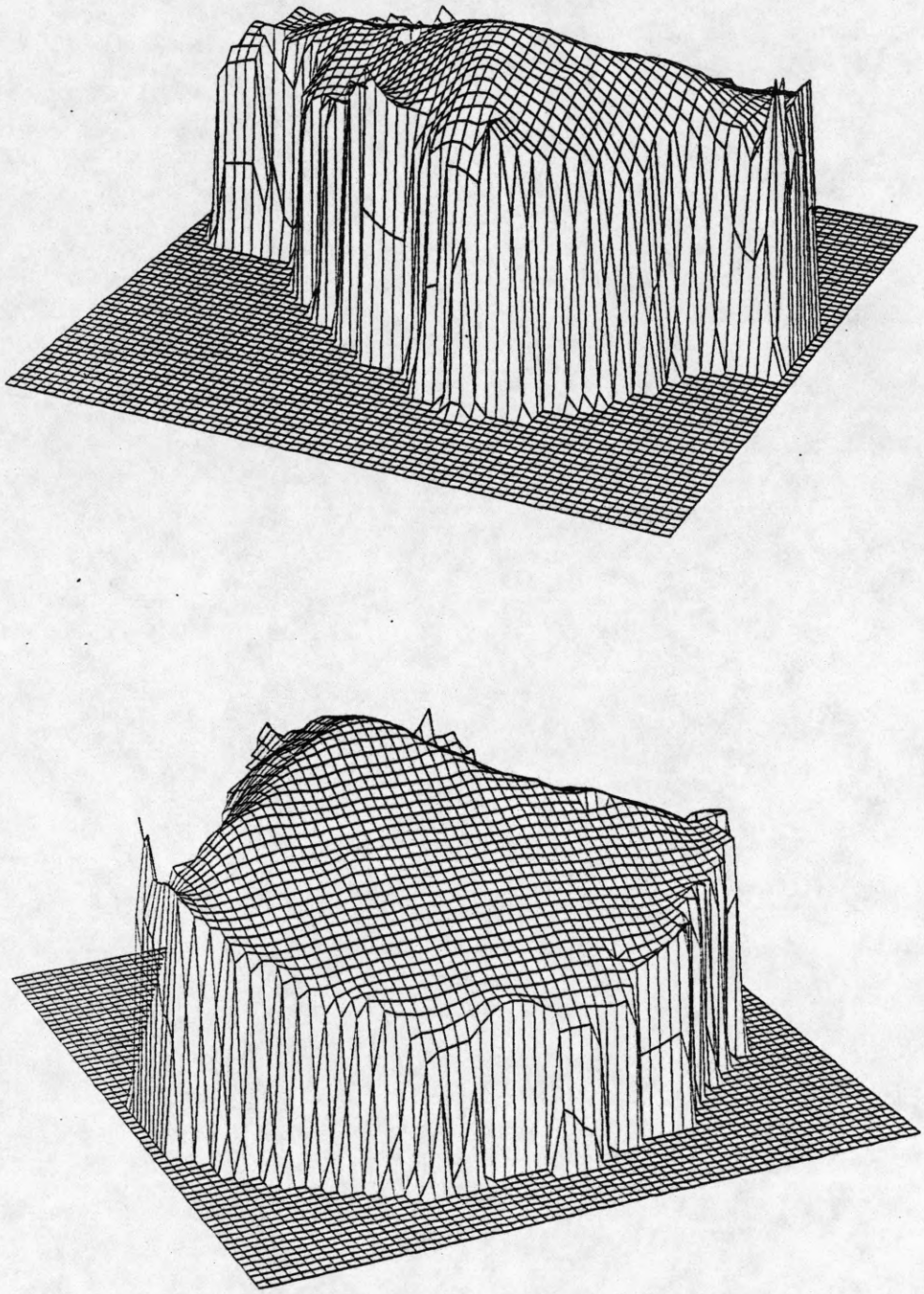

Figure 59. Reconstructed disparity surface for the sandwich image, at the 256x256 level of resolution. Disparity ranges from -36 to 22 pixels.

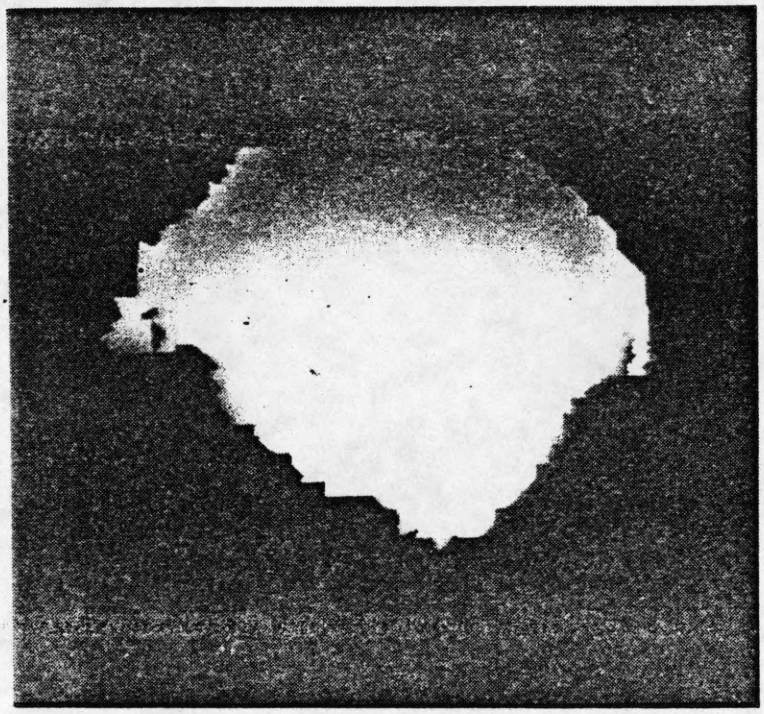

Figure 60. The 256x256 disparity surface shown as an intensity image.

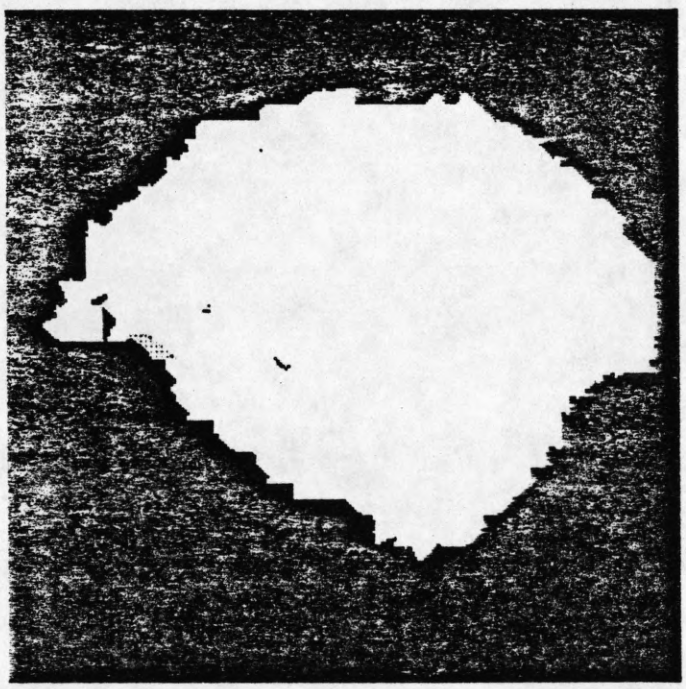

Figure 61. Status of the reconstructed 256x256surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

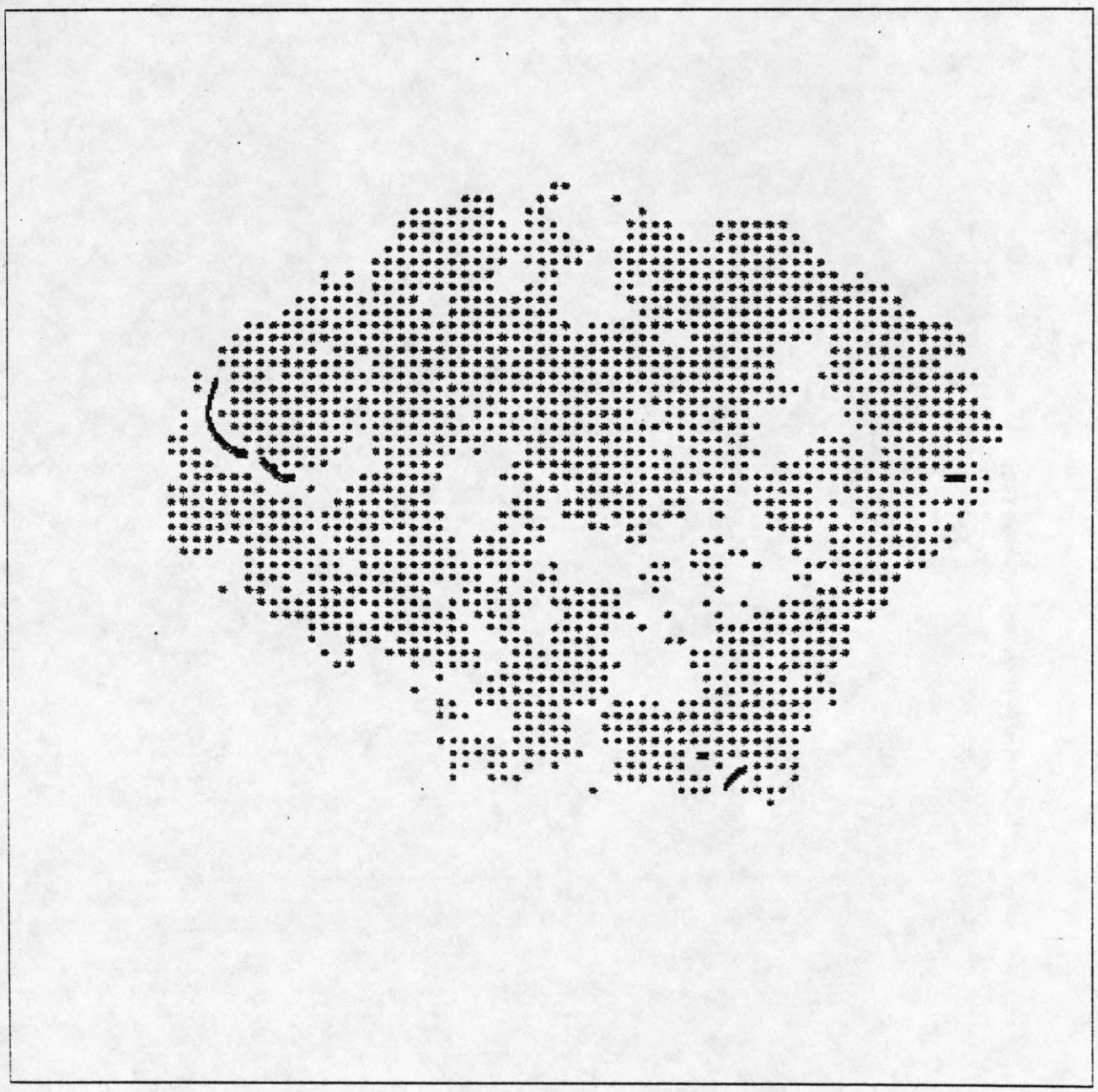

Figure 62. Quadratic patches and contours found for the sandwich image, at the 512x512 level of resolution.

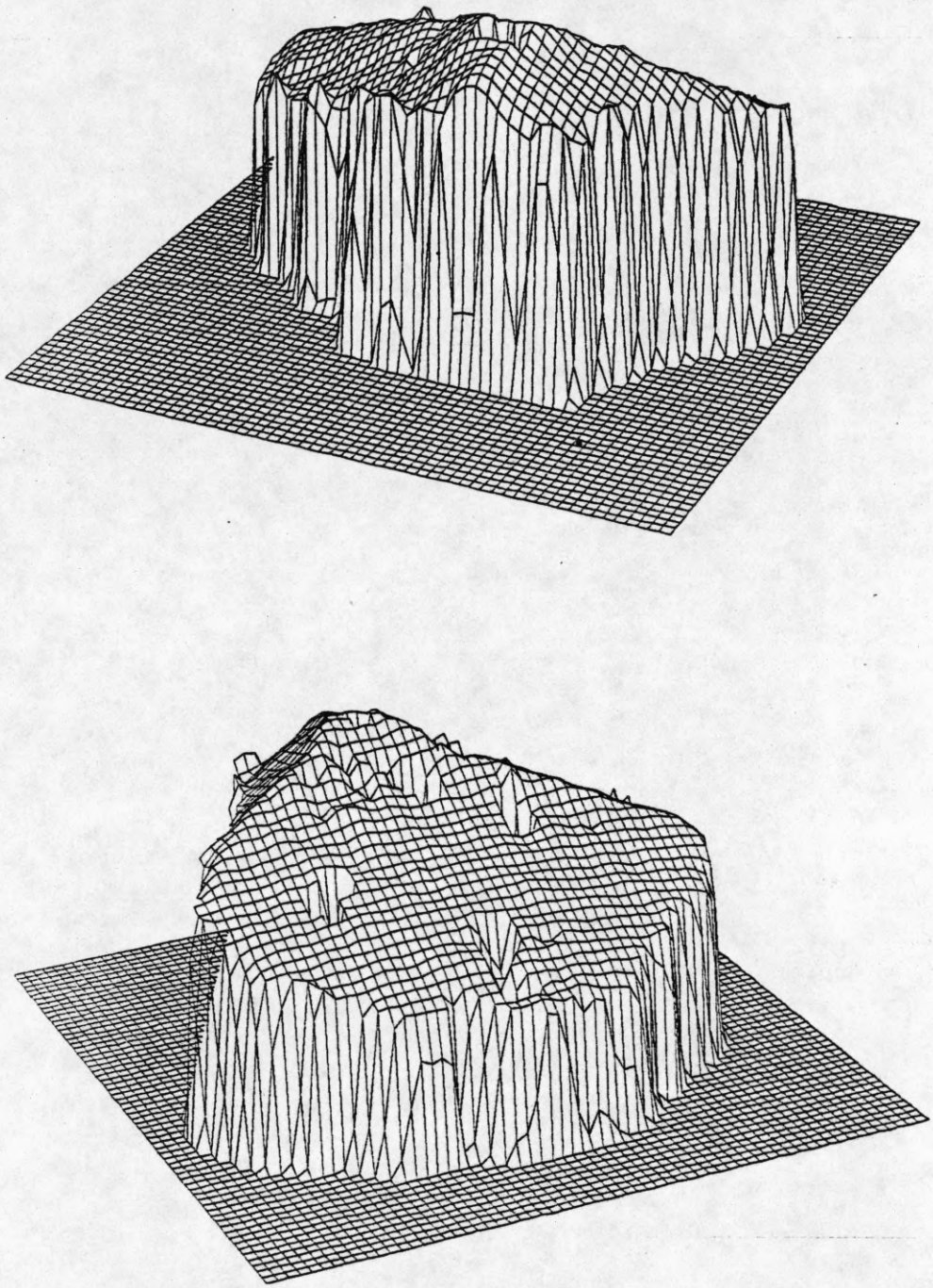

Figure 63. Reconstructed disparity surface for the sandwich image, at the 512x512 level of resolution. Disparity ranges from -58 to 45 pixels.

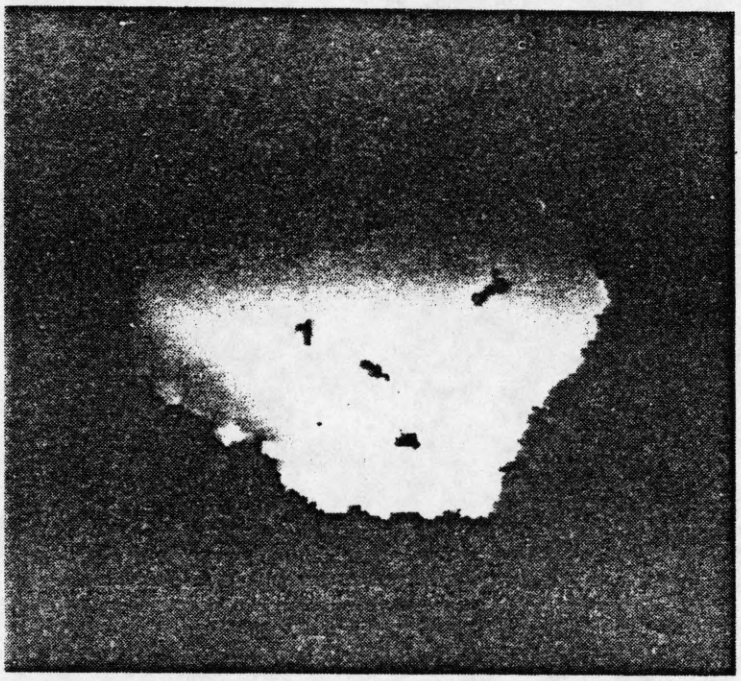

Figure 64. The 512x512 disparity surface shown as an intensity image.

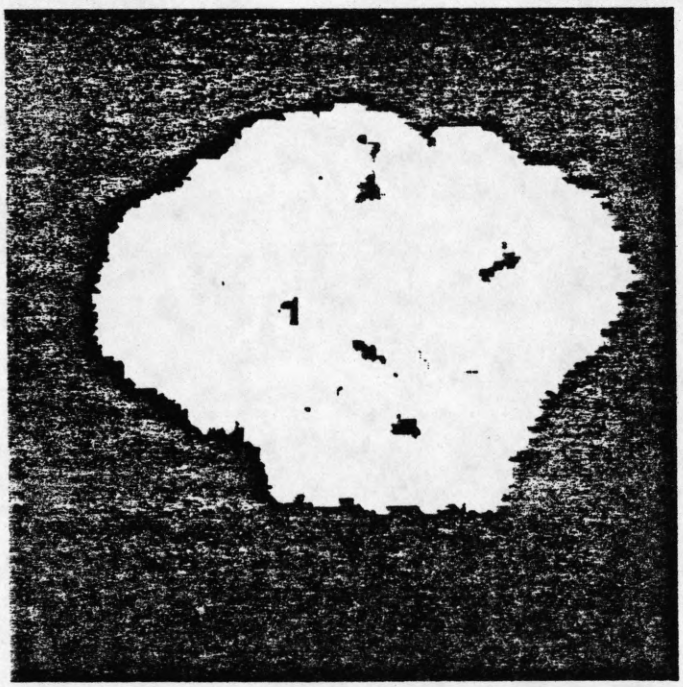

Figure 65. Status of the reconstructed 512x512 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

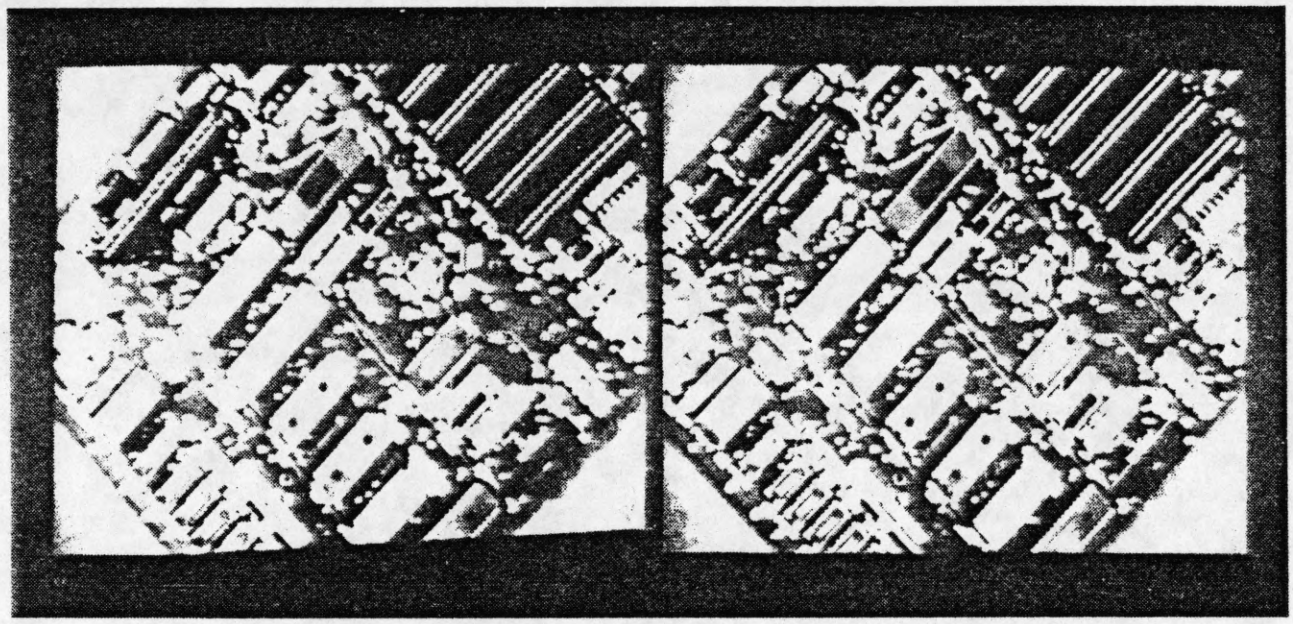

Figure 66. A 512x512 real stereo pair of images of an Apple IIe mother board. The disparity of the stereo pair ranges from about -6 pixels in the upper left corner to about 6 pixels in the lower right corner.

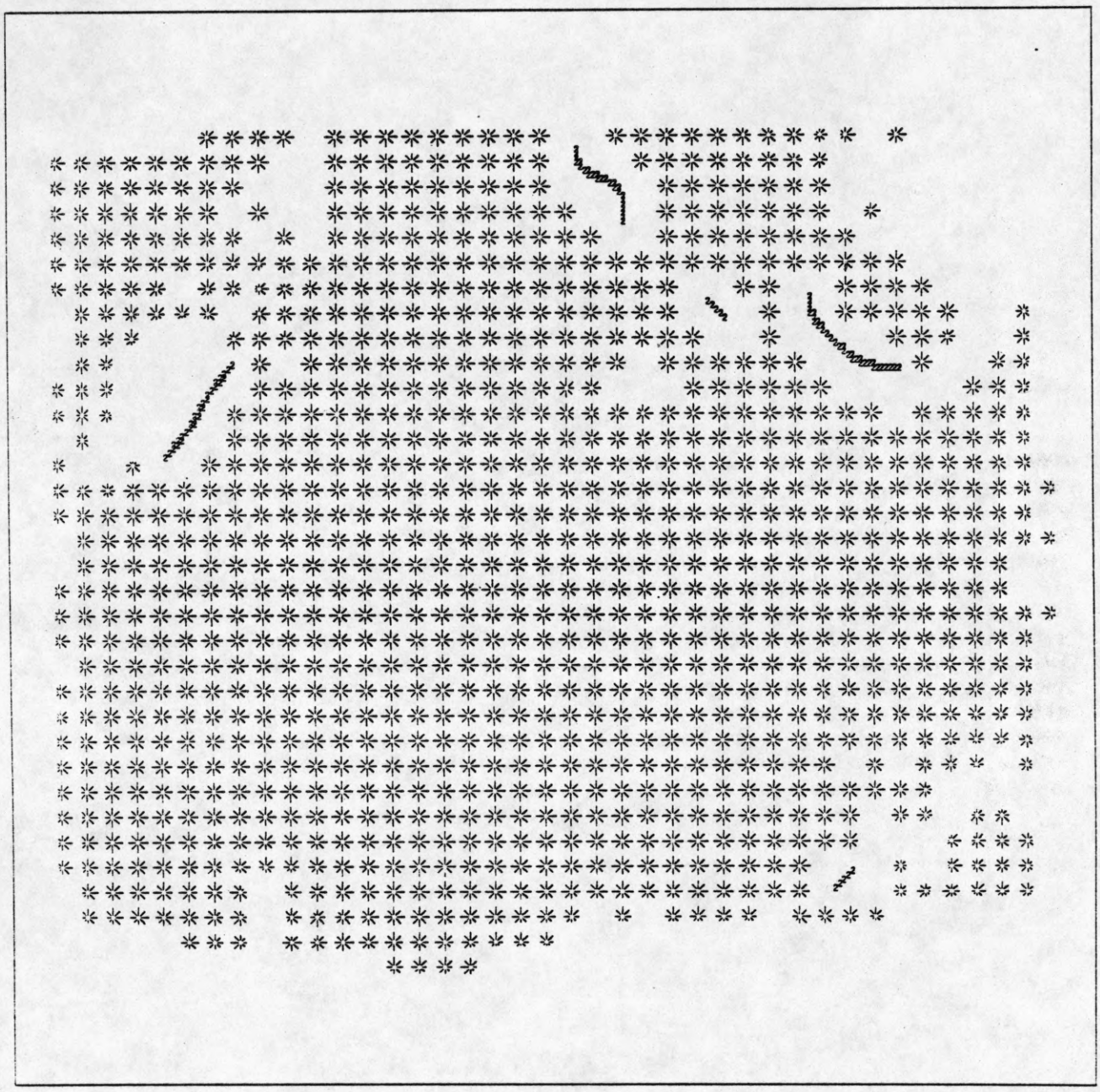

Figure 67. Quadratic patches and contours found for the apple image, at the 256x256 level of resolution.

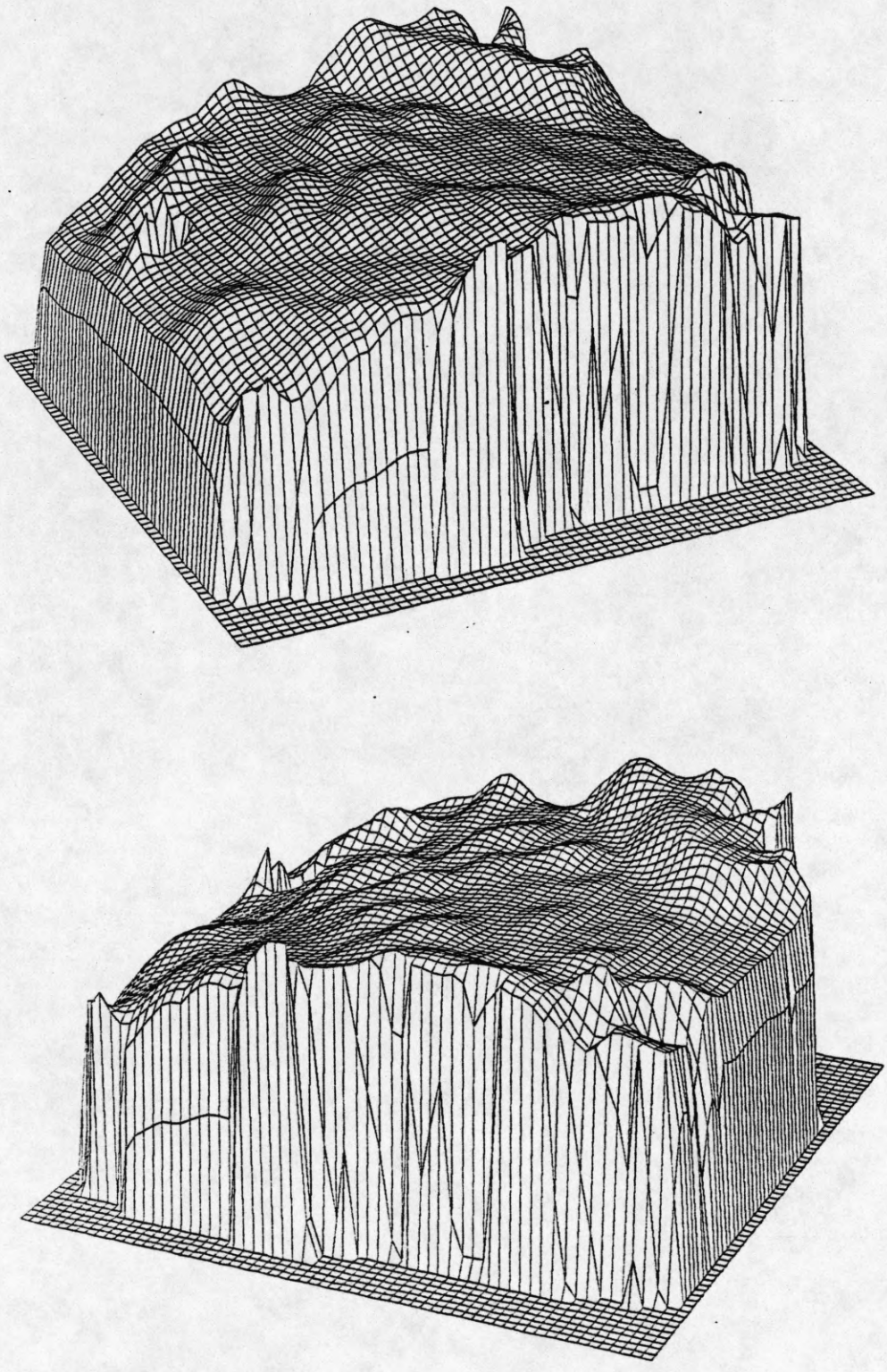

Figure 68. Reconstructed disparity surface for the apple image, at the 256x256 level of resolution. Disparity ranges from -19 to 6 pixels.

Figure 69. Quadratic patches and contours found for the apple image, at the 512x512 level of resolution.

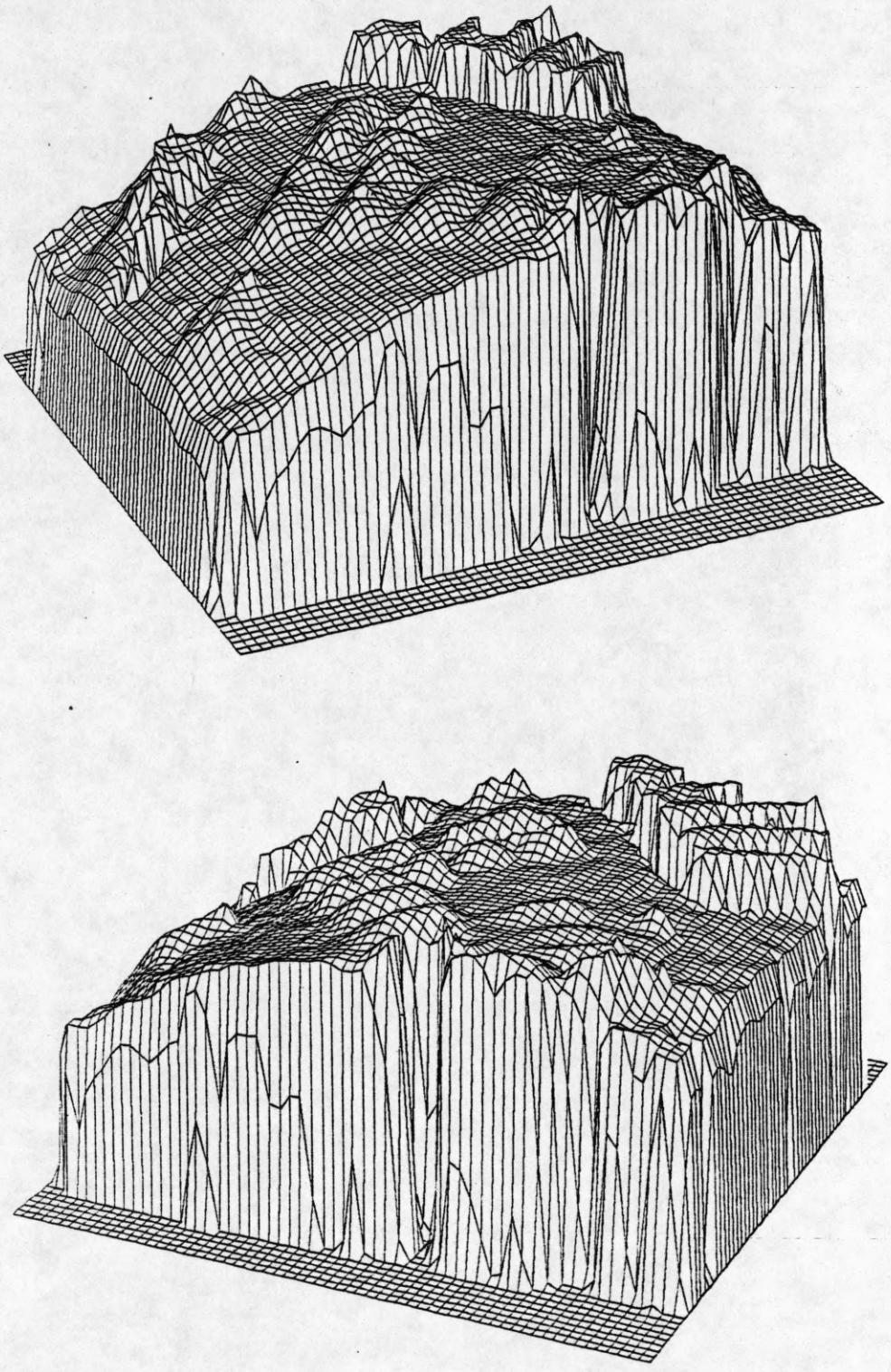

Figure 70. Reconstructed disparity surface for the apple image, at the 512x512 level of resolution. Disparity ranges from -34 to 11 pixels.

Figure 71. The 512x512 disparity surface shown as an intensity image.

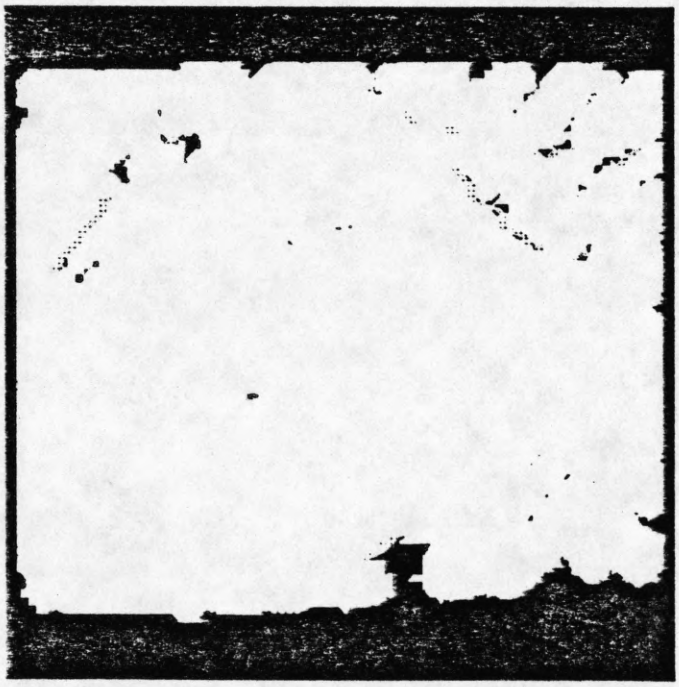

Figure 72. Status of the reconstructed 512x512 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

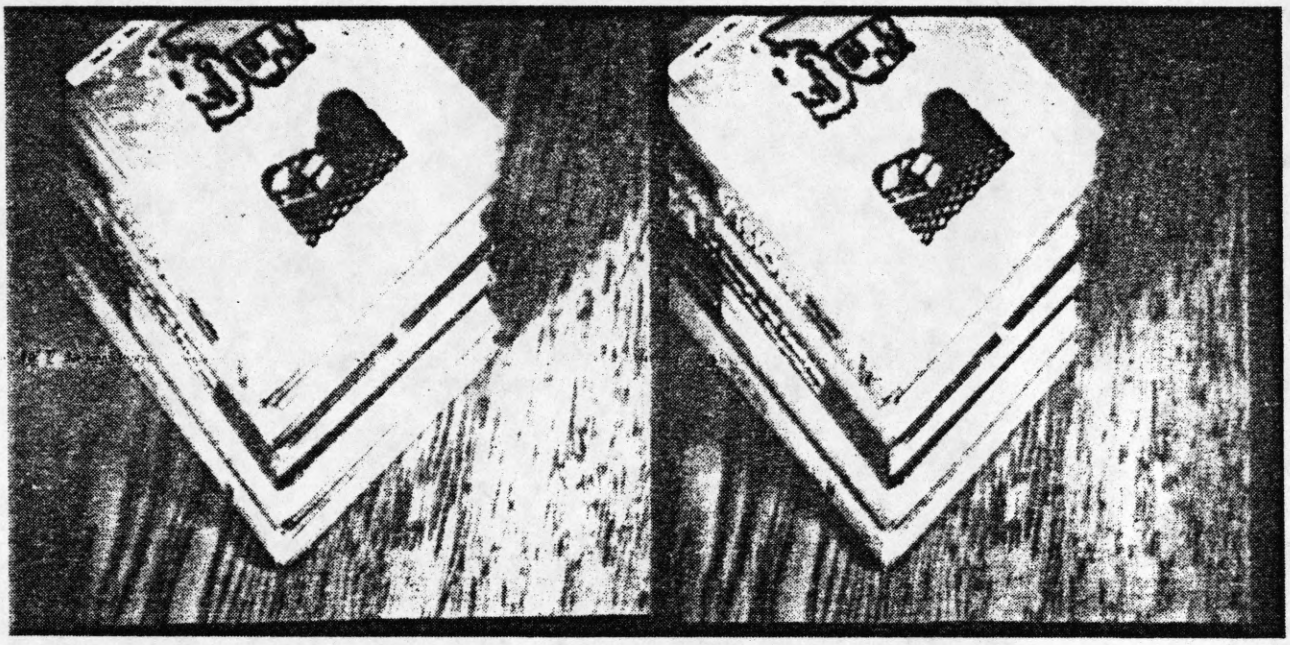

Figure 73. A 512x512 real stereo pair of images of a pile of books. The disparity of the stereo pair ranges from about -30 pixels for the farthest point of the background to about 16 pixels for the closest point of the books.

Figure 74. Quadratic patches and contours found for the books image, at the 256x256 level of resolution.

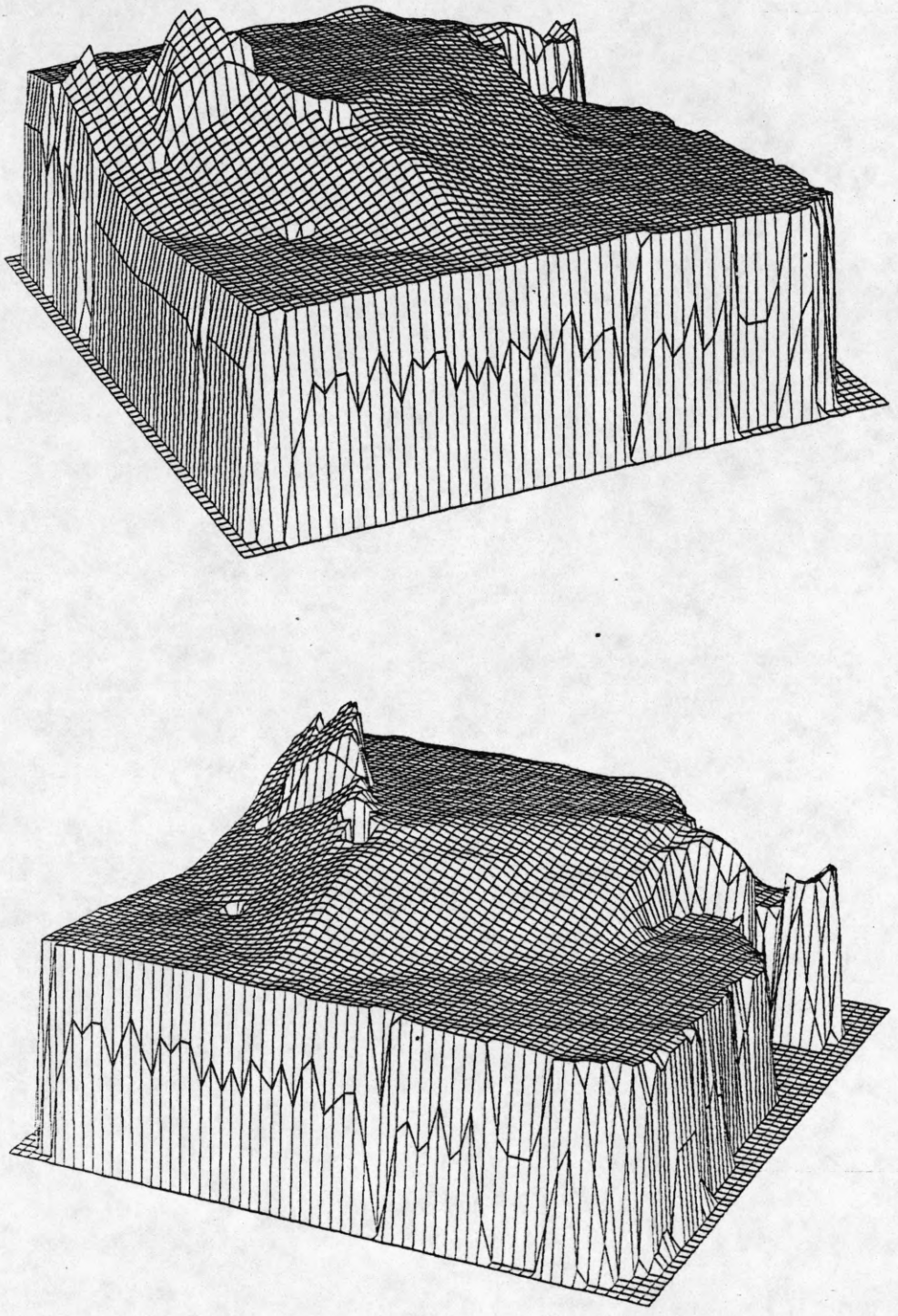

Figure 75. Reconstructed disparity surface for the books image, at the 256x256 level of resolution. Disparity ranges from -46 to 21 pixels.

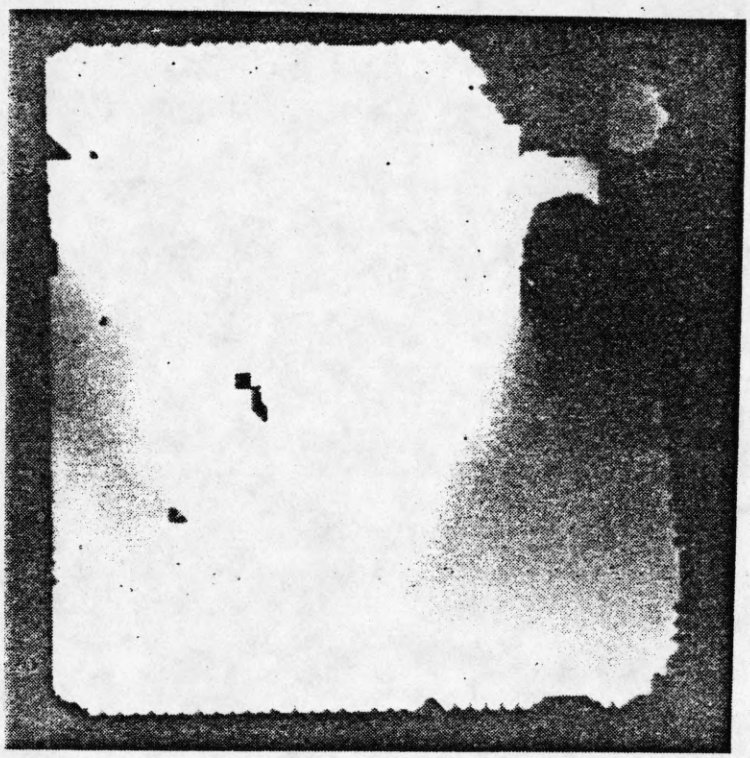

Figure 76. The 256x256 disparity surface shown as an intensity image.

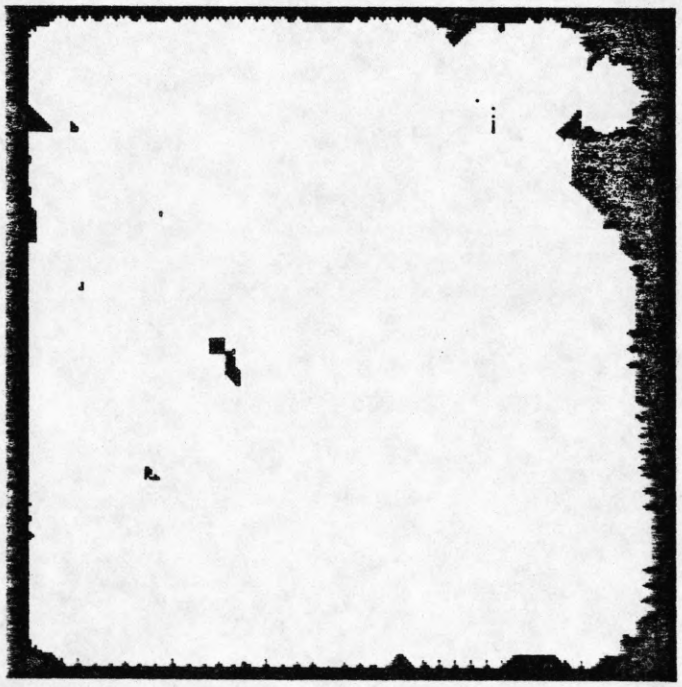

Figure 77. Status of the reconstructed 256x256 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

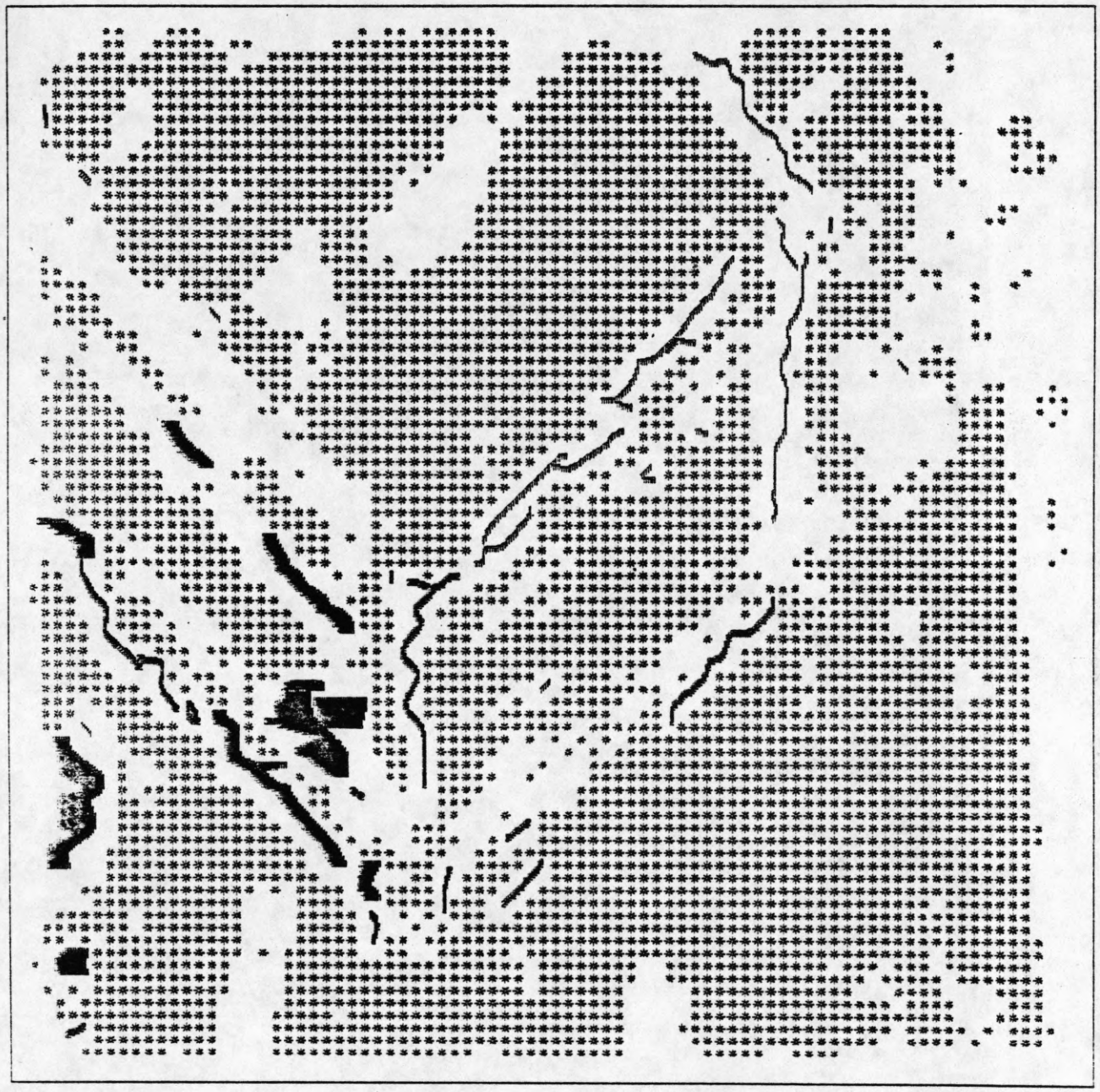

Figure 78. Quadratic patches and contours found for the books image, at the 512x512 level of resolution.

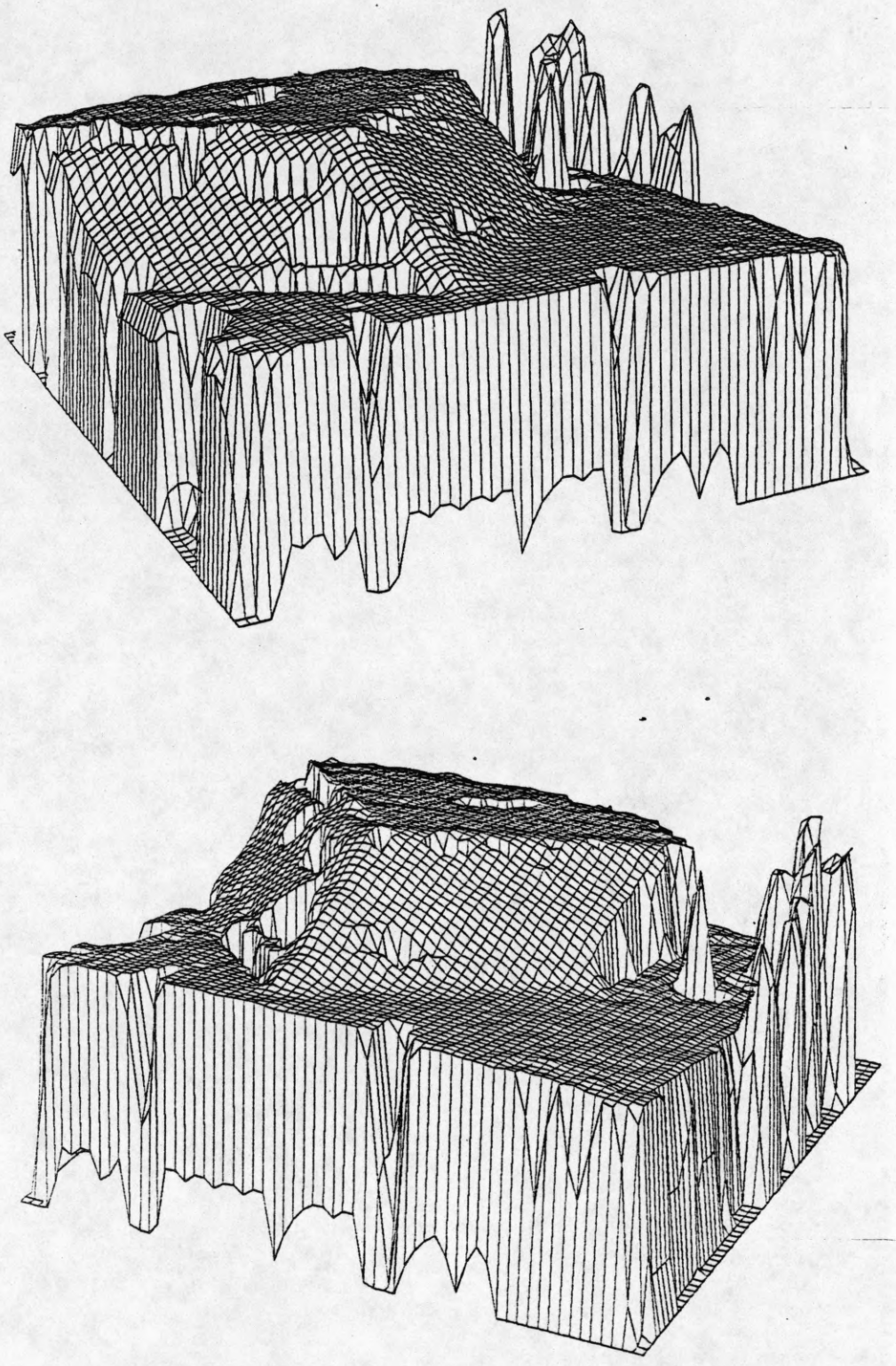

Figure 79. Reconstructed disparity surface for the books image, at the 512x512 level of resolution. Disparity ranges from -74 to 18 pixels.

Figure 80. The 512x512 disparity surface shown as an intensity image.

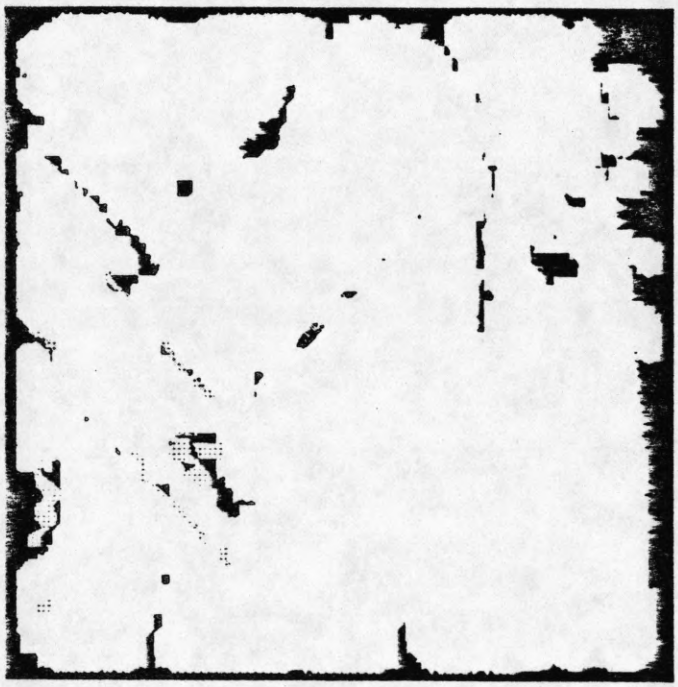

Figure 81. Status of the reconstructed 512x512 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

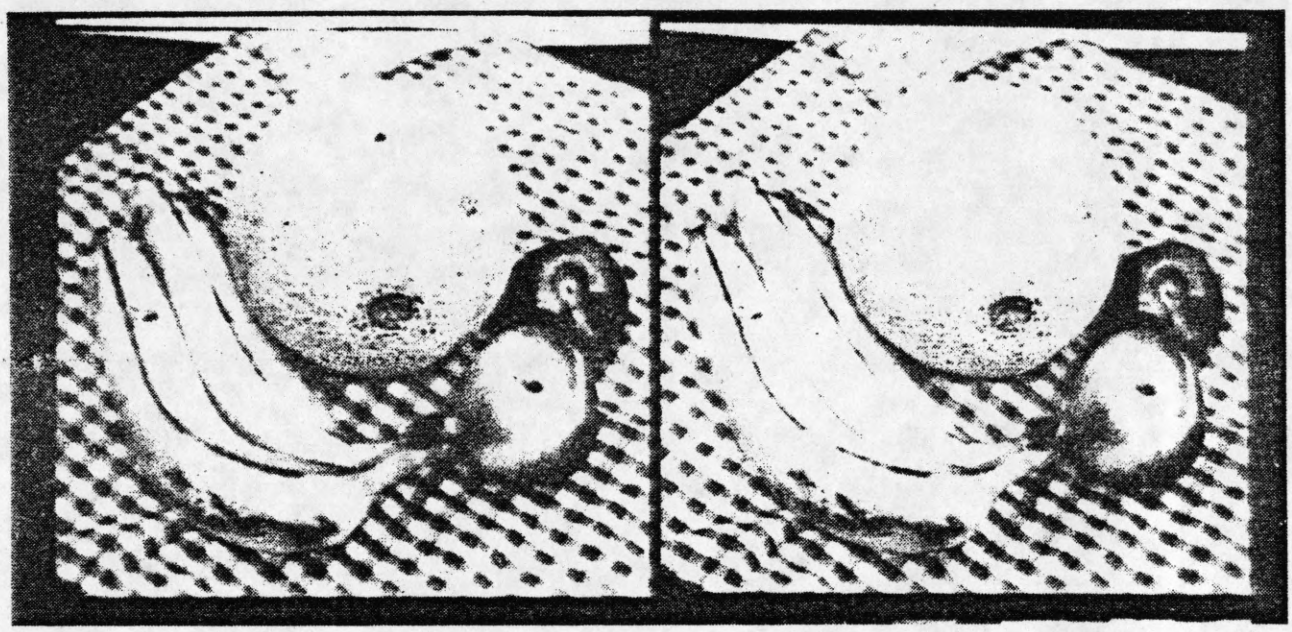

Figure 82. A 512x512 real stereo pair of images of some fruit. The disparity of the stereo pair ranges from about -26 pixels at the top of the image to about 13 pixels at the bottom of the image.

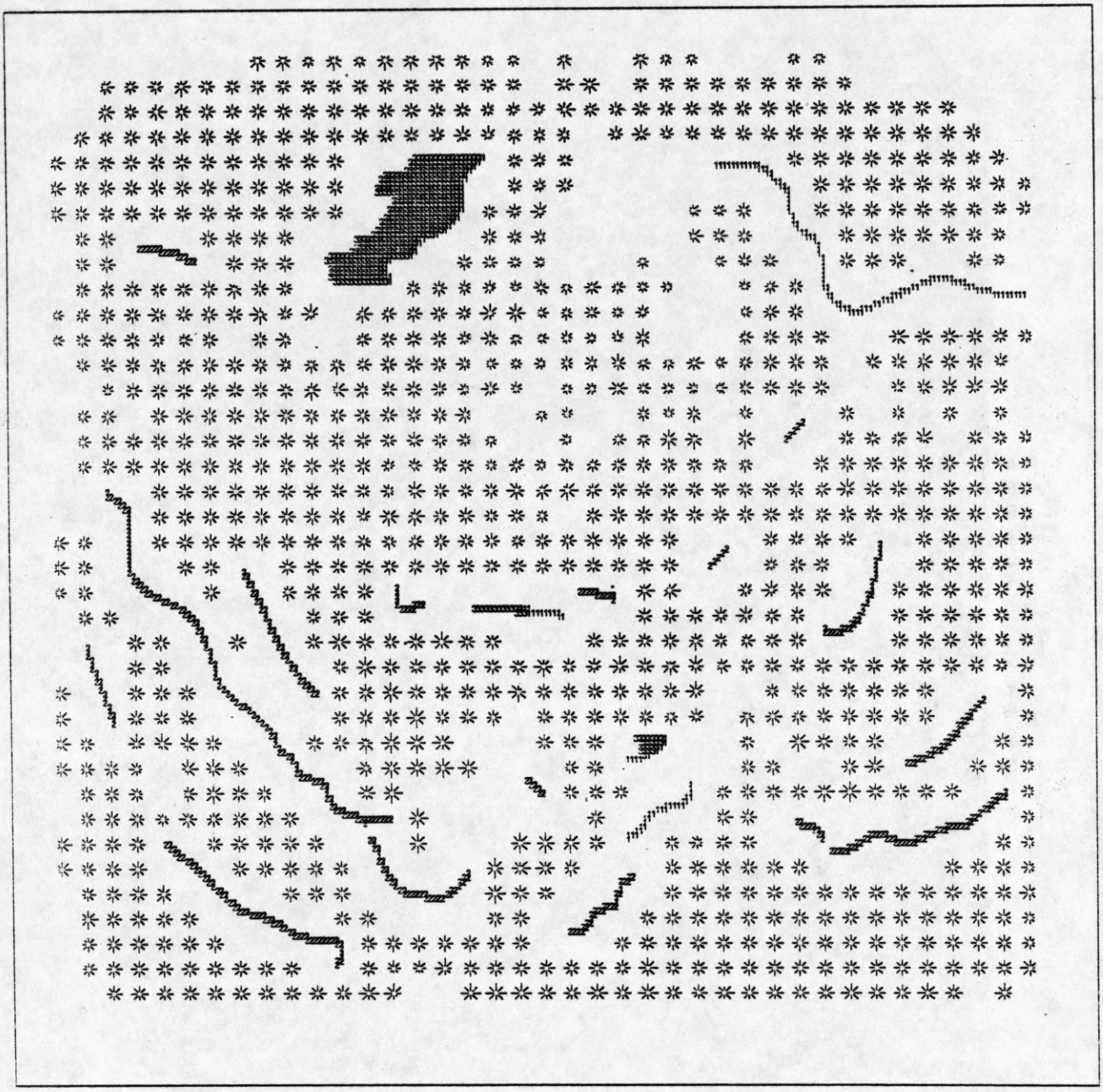

Figure 83. Quadratic patches and contours found for the fruit image, at the 256x256 level of resolution.

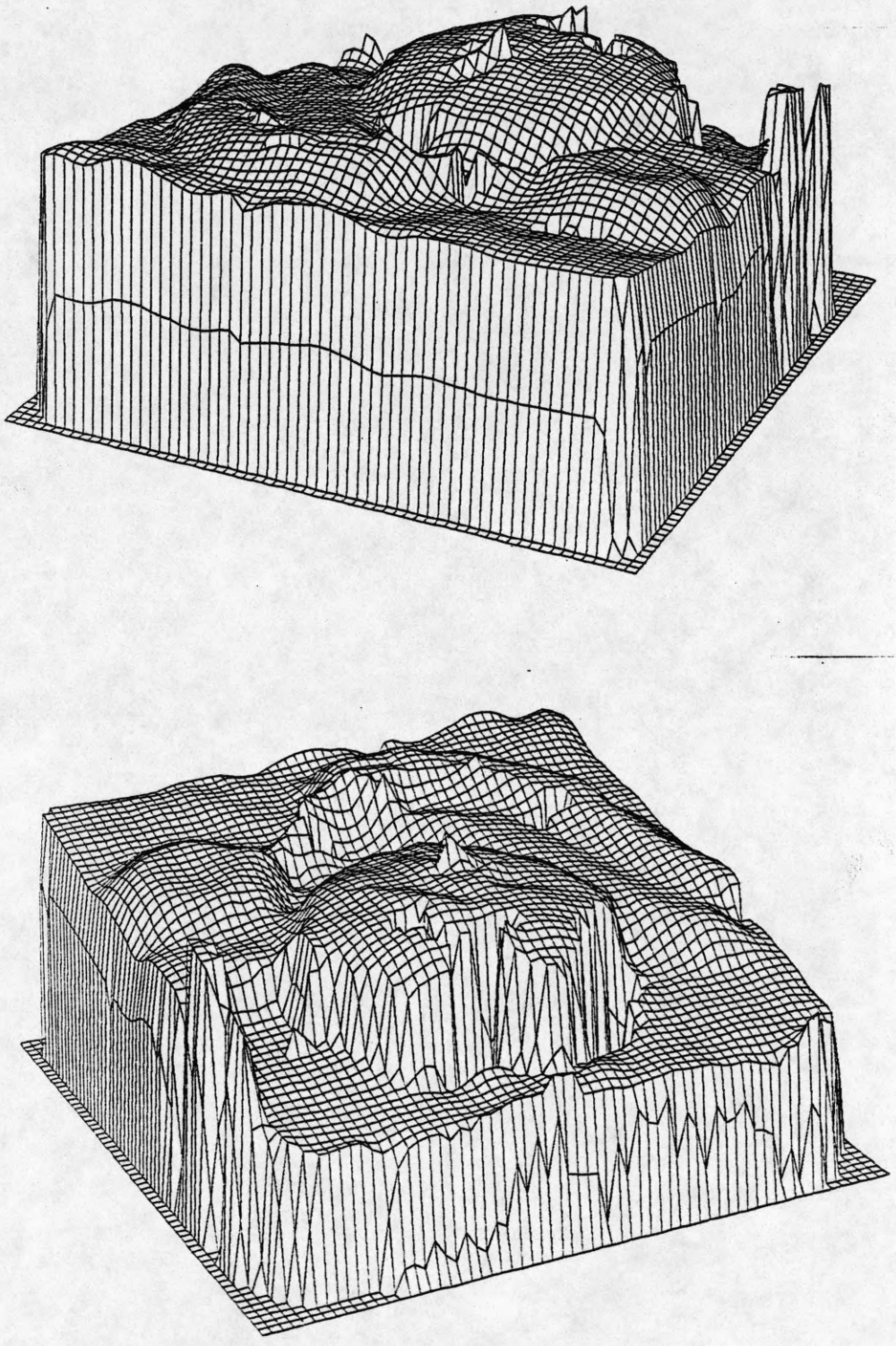

Figure 84. Reconstructed disparity surface for the fruit image, at the 256x256 level of resolution. Disparity ranges from -36 to 15 pixels.

Figure 85. The 256x256 disparity surface shown as an intensity image.

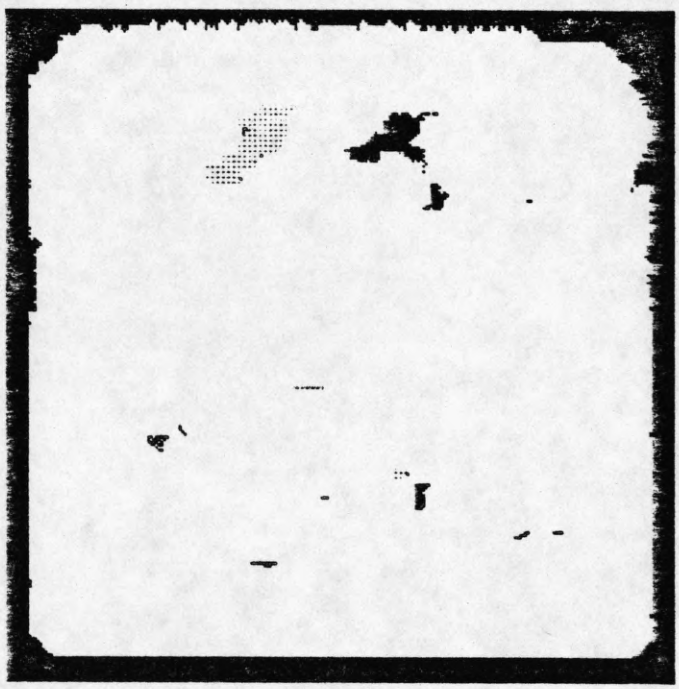

Figure 86. Status of the reconstructed 256x256 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

Figure 87. Quadratic patches and contours found for the fruit image, at the 512x512 level of resolution.

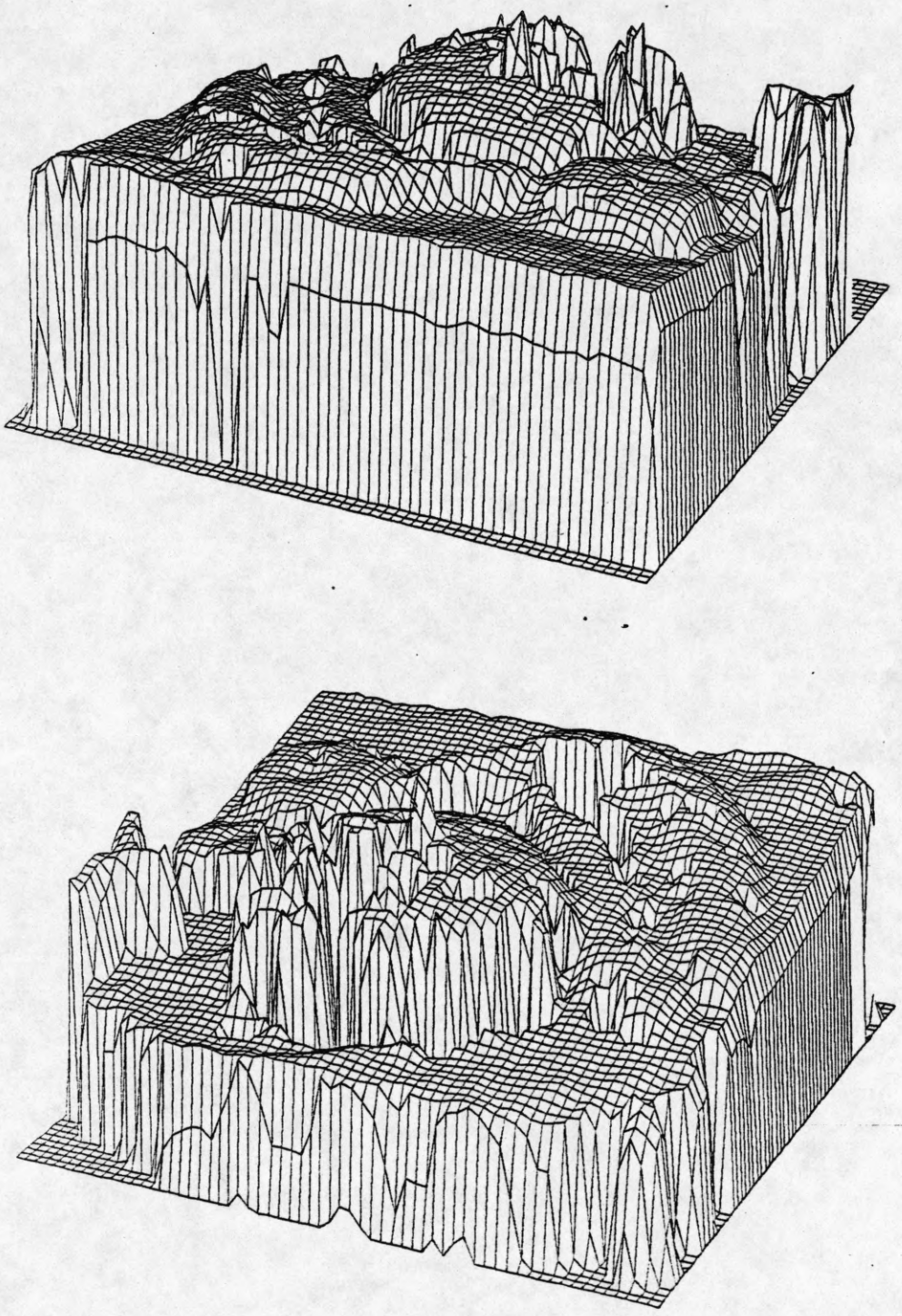

Figure 88. Reconstructed disparity surface for the fruit image, at the 512x512 level of resolution. Disparity ranges from -76 to 36 pixels.

Figure 89. The 512x512 disparity surface shown as an intensity image.

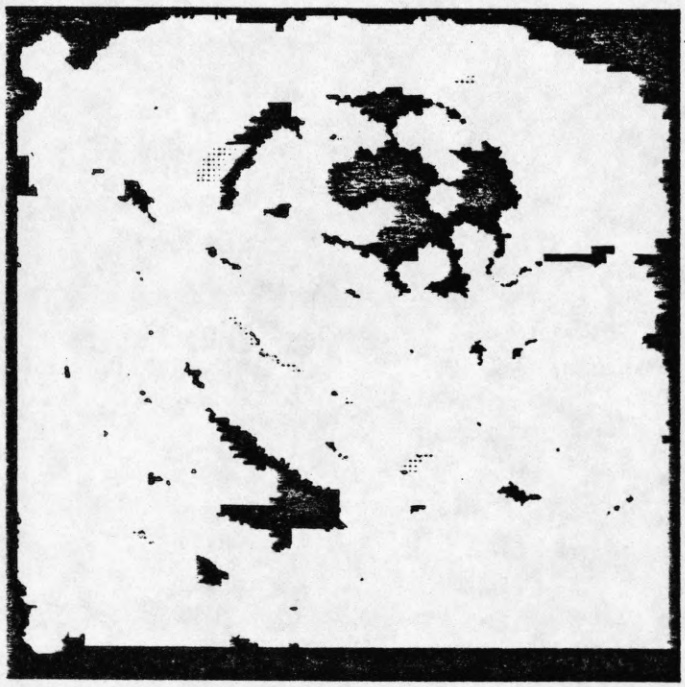

Figure 90. Status of the reconstructed 512x512 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

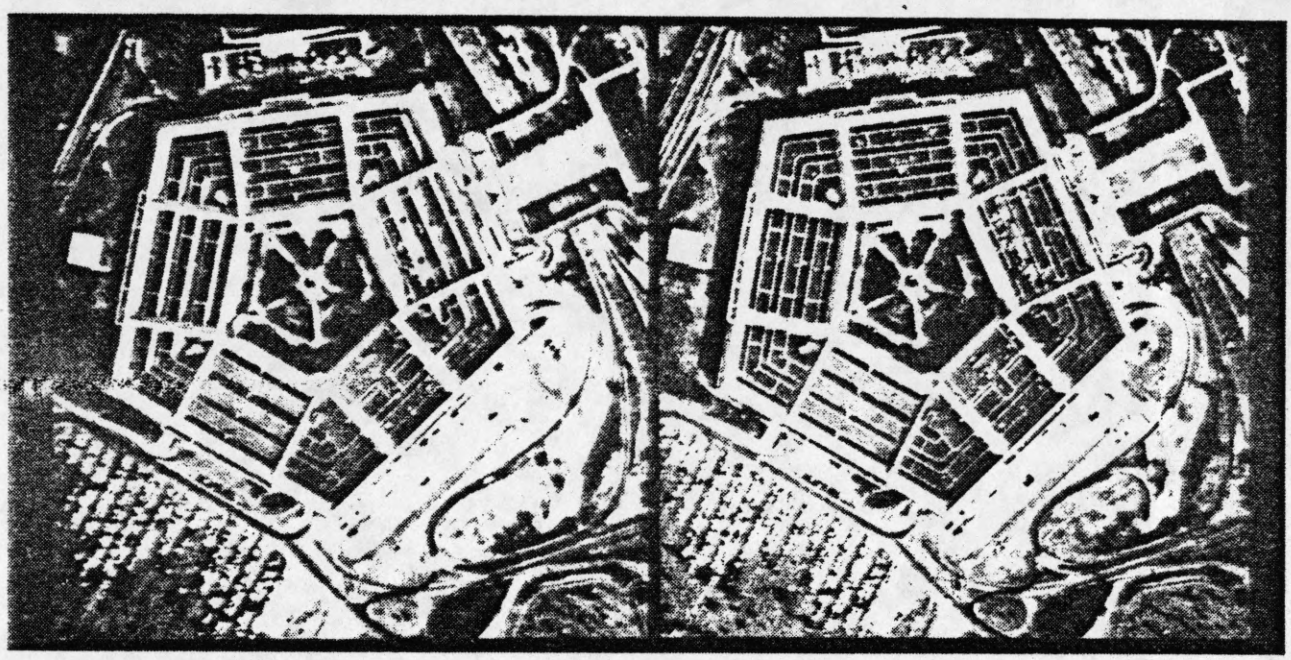

Figure 91. A 512x512 real stereo pair of an aerial view of the Pentagon Building. The disparity of the backgournd is about -5 pixels and the disparity of the top of the building is about 5 pixels.

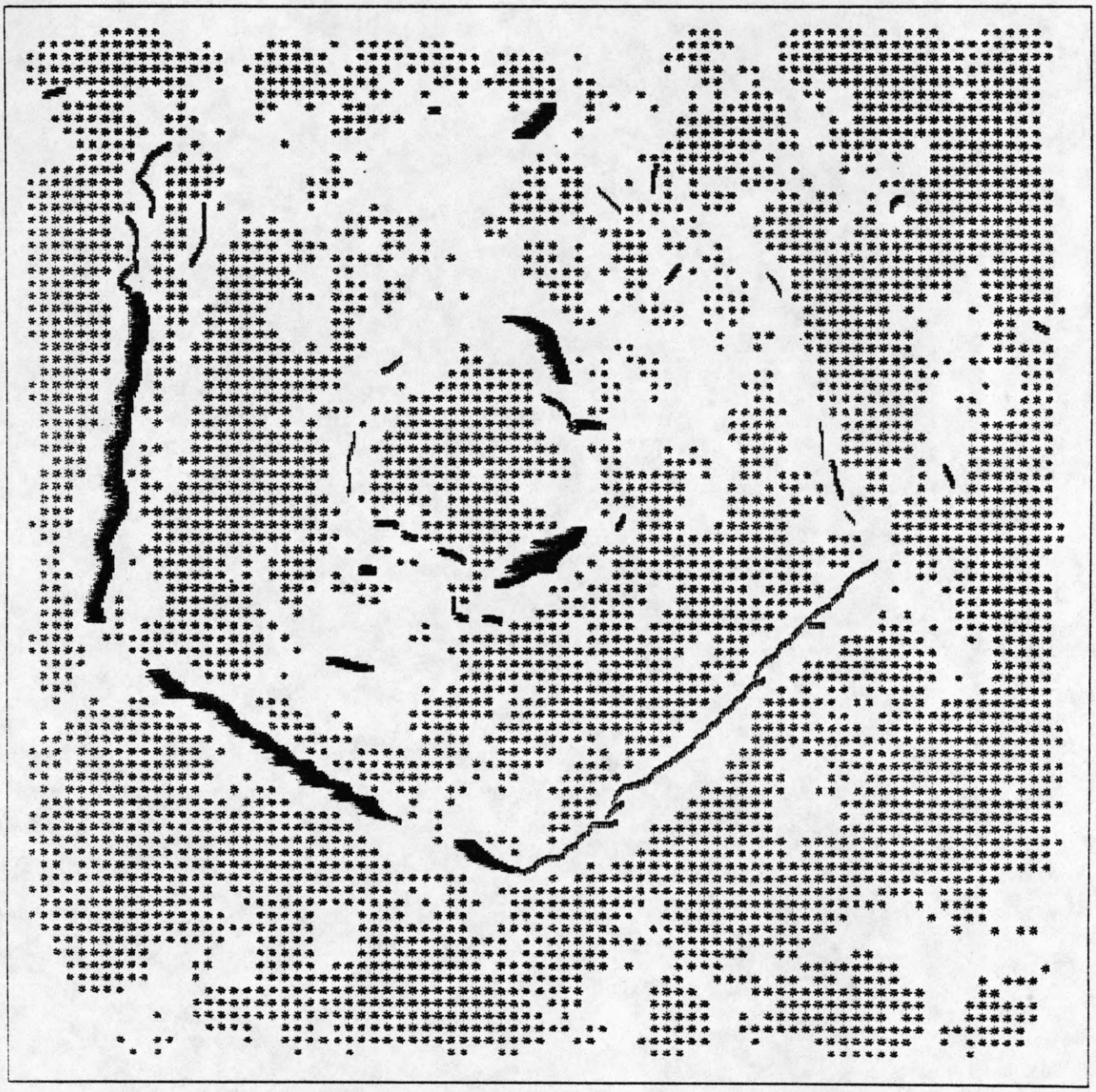

Figure 92. Quadratic patches and contours found for the pentagon image, at the 512x512 level of resolution.

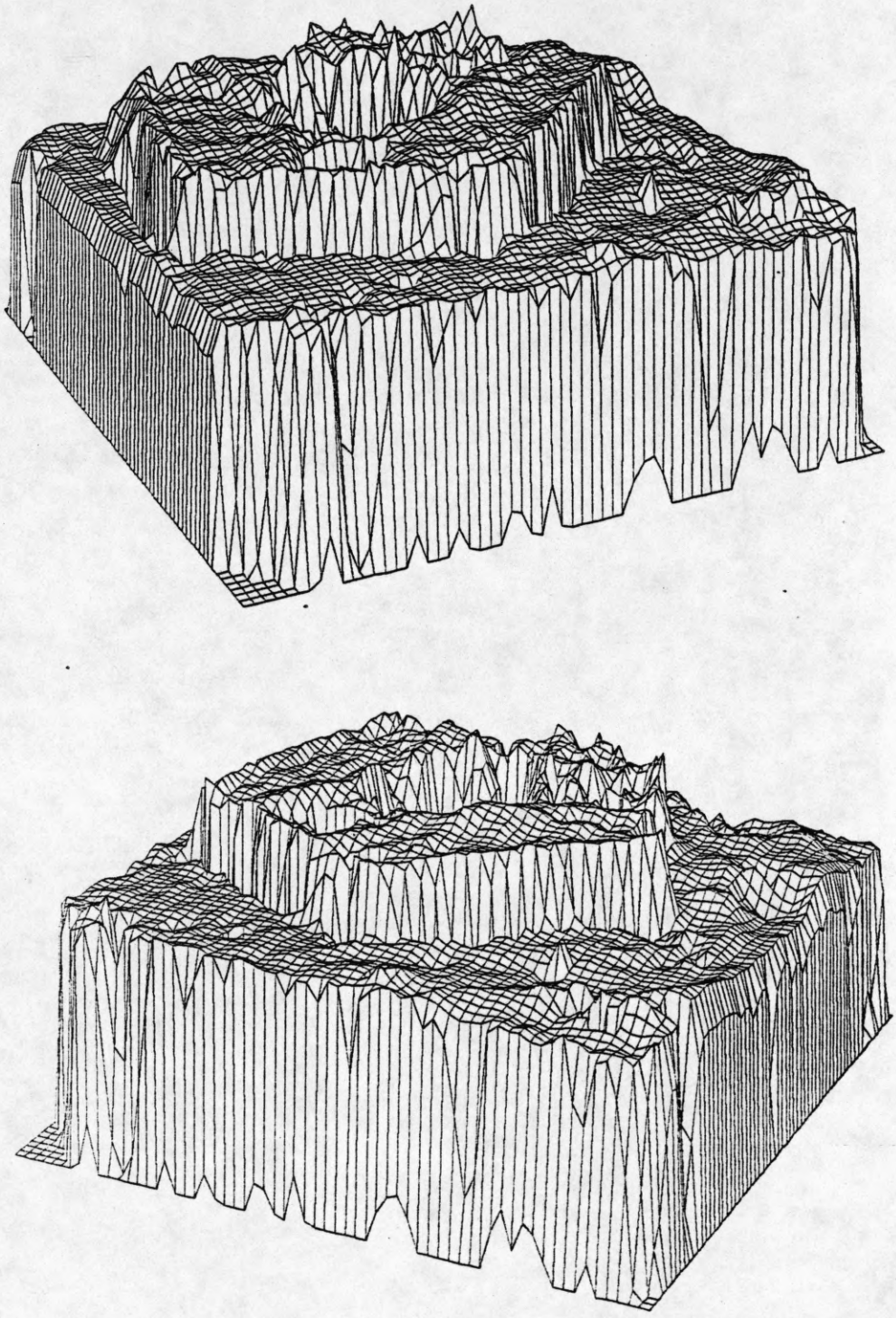

Figure 93. Reconstructed disparity surface for the pentagon image, at the 512x512 level of resolution. Disparity ranges from -30 to 9 pixels.

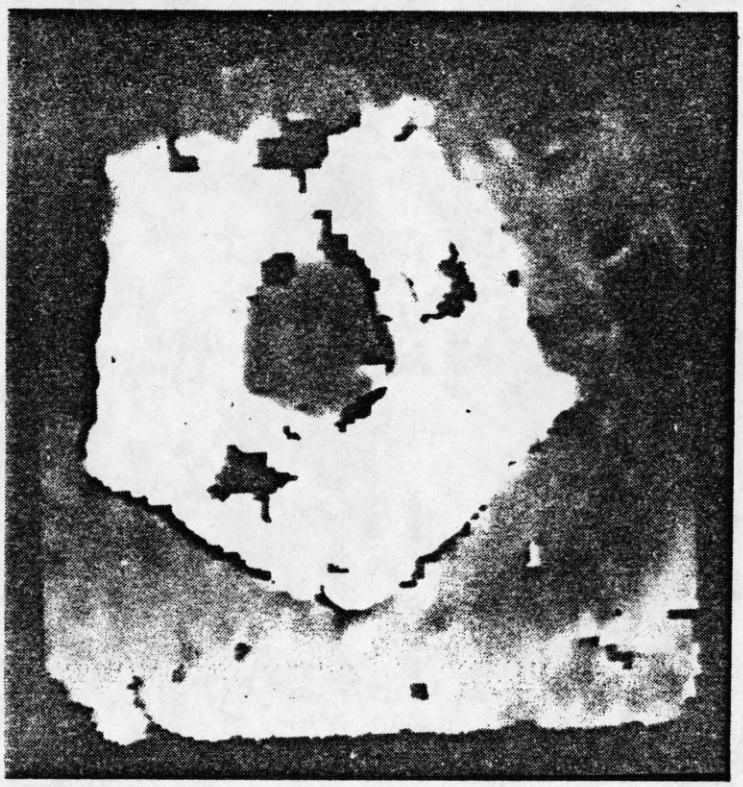

Figure 94. The 512x512 disparity surface shown as an intensity image.

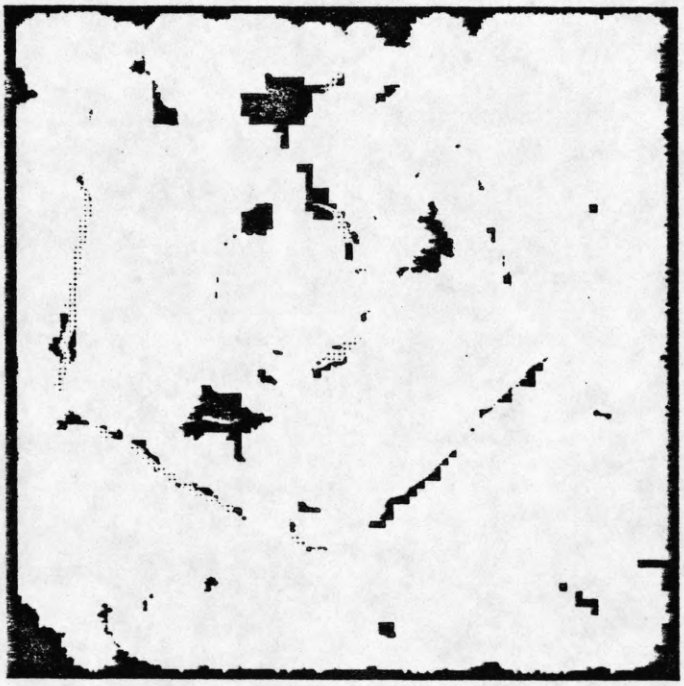

Figure 95. Status of the reconstructed 512x512 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

Figure 96. A 512x512 real stereo pair of images of a Renault auto part. The disparity of the auto part ranges from about 6 pixels at the leftmost tip to about 30 pixels at the bottom.

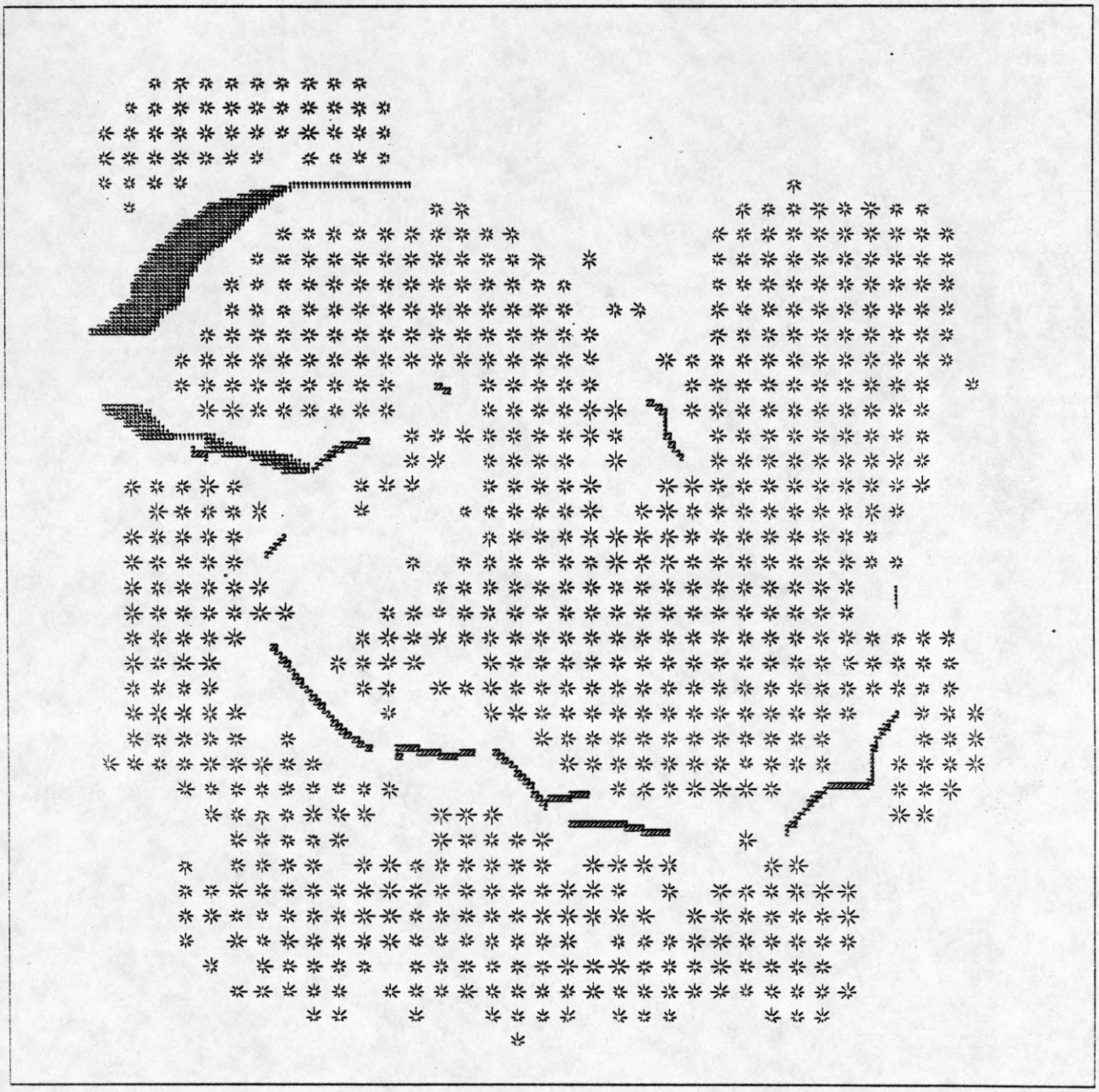

Figure 97. Quadratic patches and contours found for the renault image, at the 256x256 level of resolution.

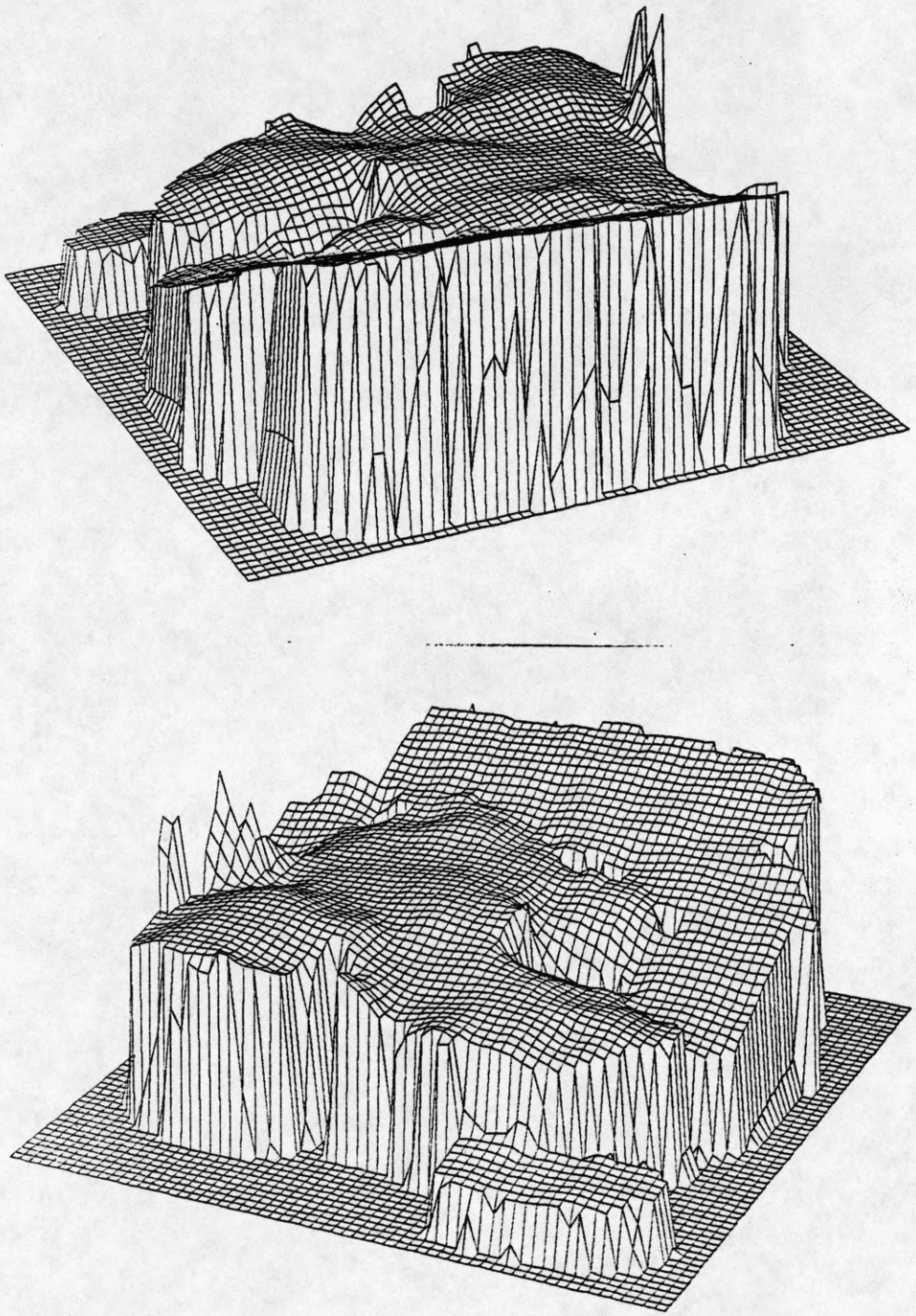

Figure 98. Reconstructed disparity surface for the renault image, at the 256x256 level of resolution. Disparity ranges from -25 to 31 pixels.

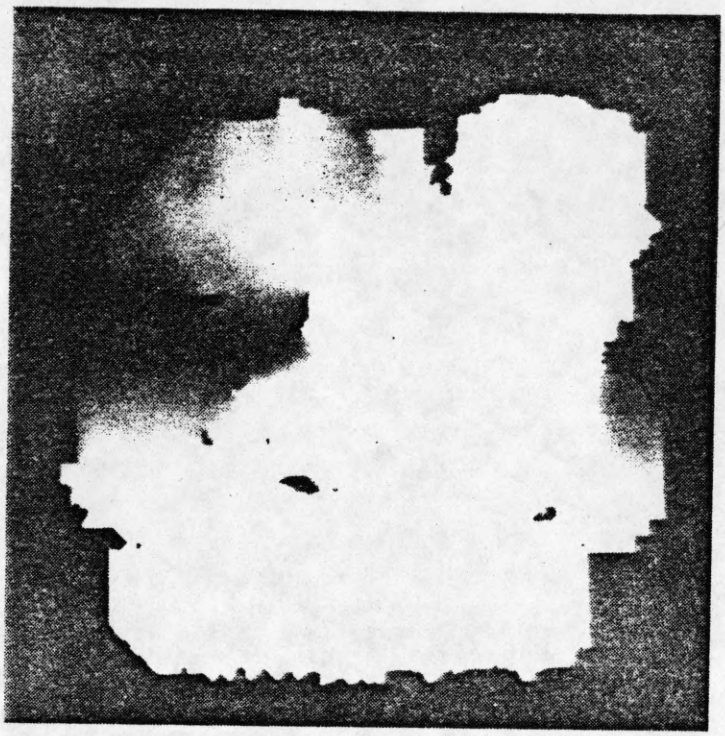

Figure 99. The 256x256 disparity surface shown as an intensity image.

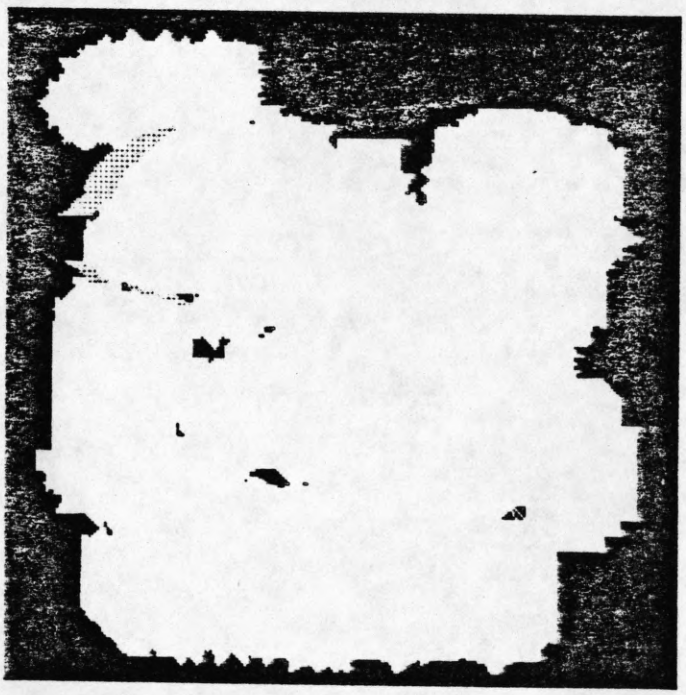

Figure 100. Status of the reconstructed 256x256 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

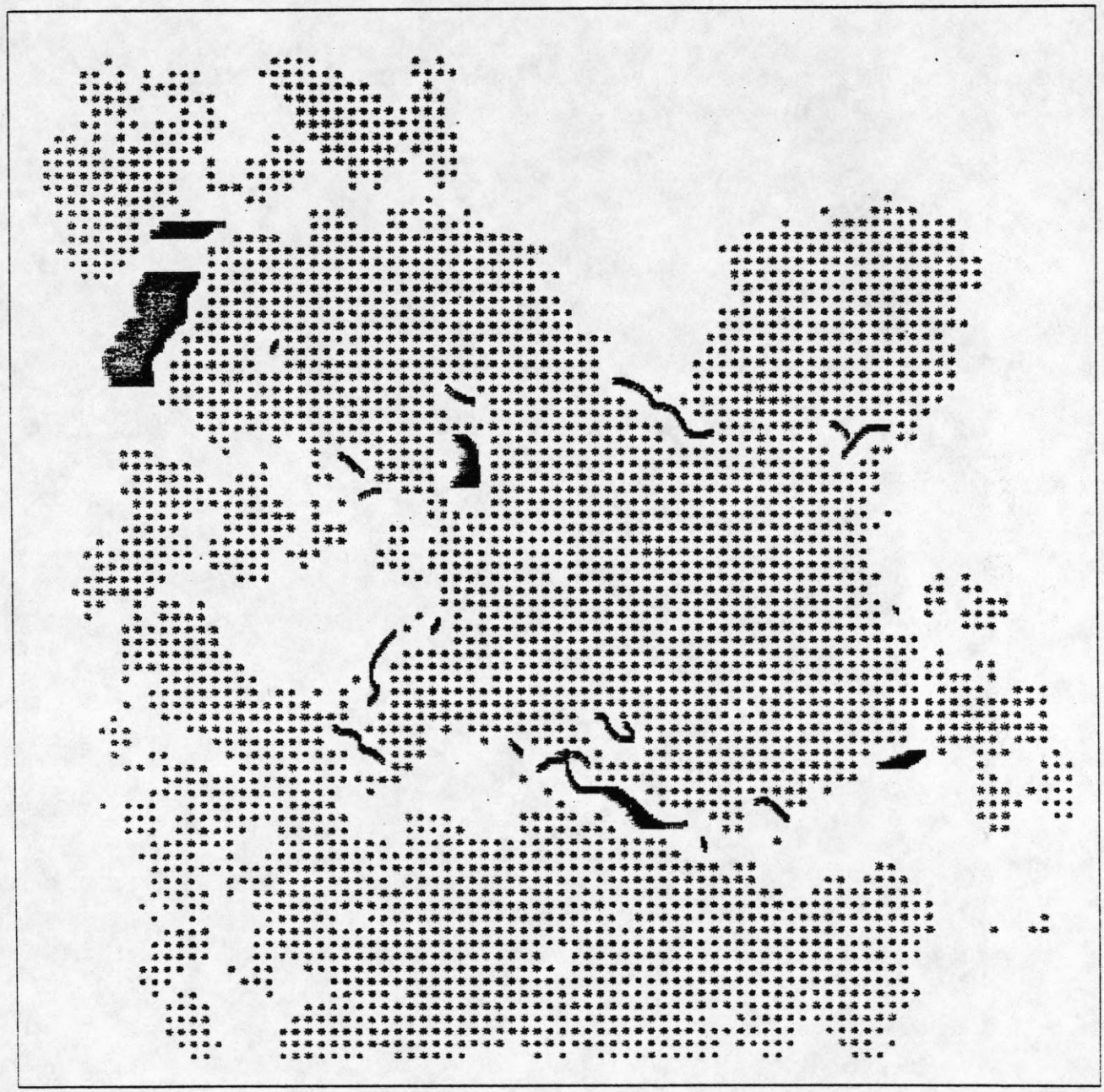

Figure 101. Quadratic patches and contours found for the renault image, at the 512x512 level of resolution.

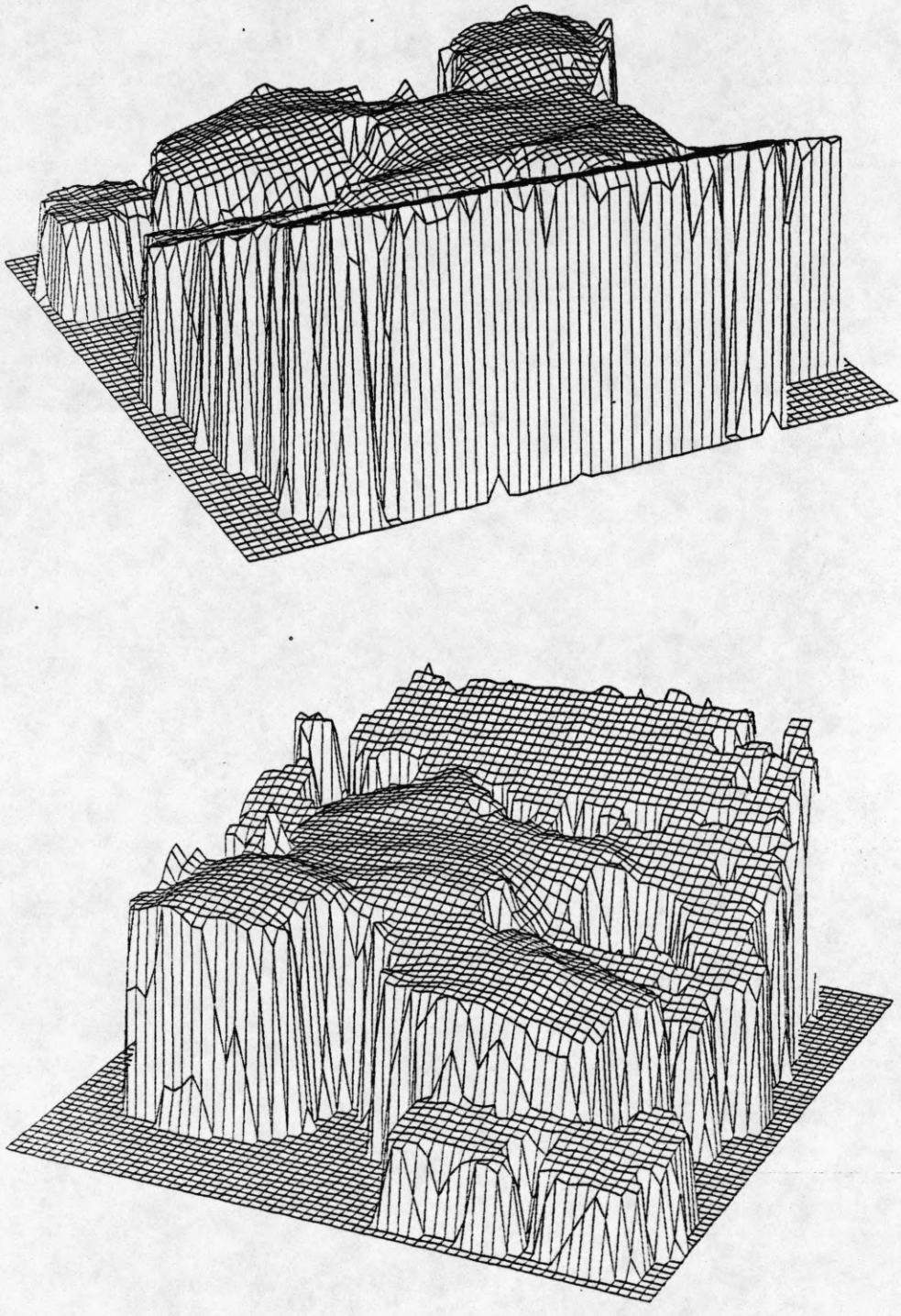

Figure 102. Reconstructed disparity surface for the renault image, at the 512x512 level of resolution. Disparity ranges from -60to 57 pixels.

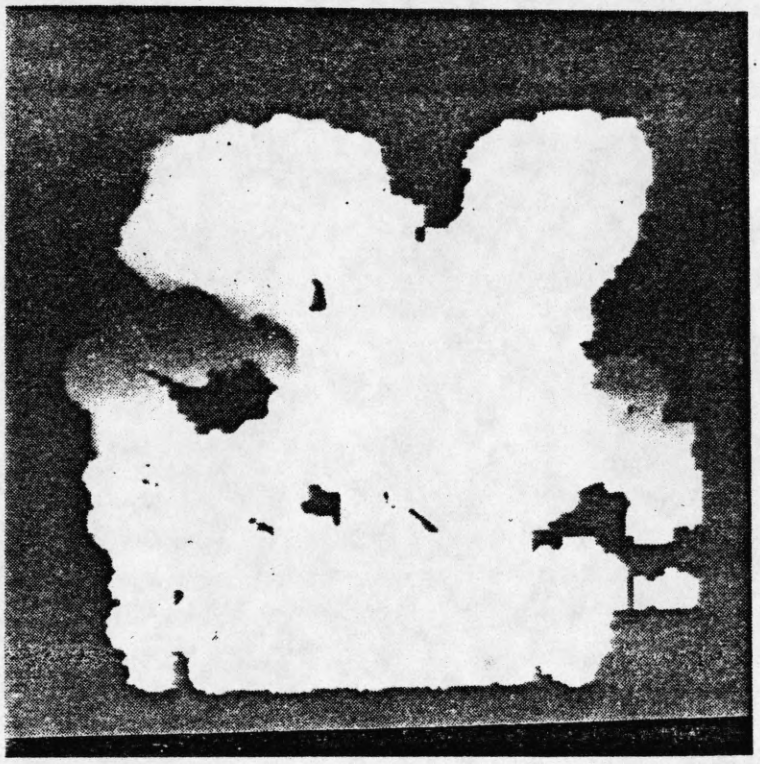

Figure 103. The 512x512 disparity surface shown as an intensity image.

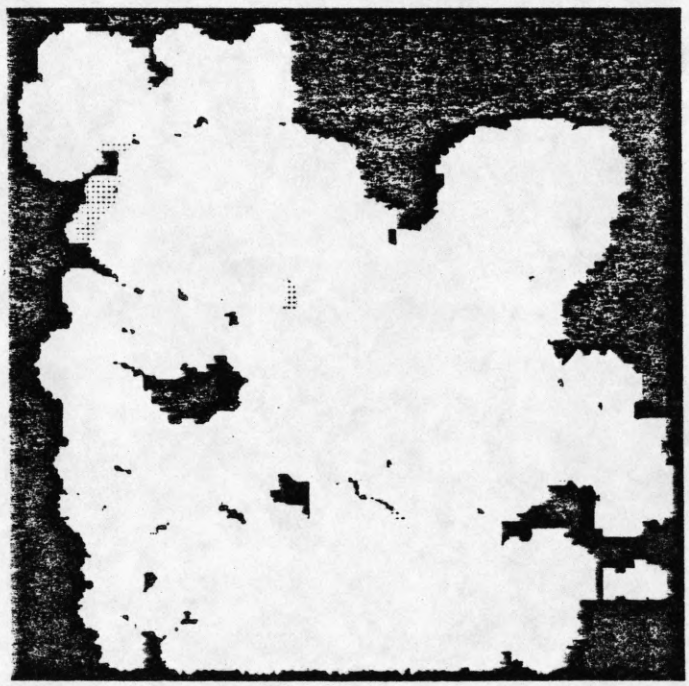

Figure 104. Status of the reconstructed 512x512 surface: black indicates "unknown" areas, gray indicates "occluded" areas, and white indicates "known" areas.

6. CONCLUSIONS

Existing stereo algorithms complete the matching process before interpolating to obtain a dense depth map. Uniqueness of matching is enforced by conditions that involve simple relationships among local disparity values in the image. Since each disparity value implies a depth value, a stereo pair with ambiguous matches implies multiple surfaces which exhibit different smoothness properties. We expect objects in the real word to have smooth surfaces, in the sense that the normal direction varies slowly except across occluding and ridge boundaries. Therefore the matching process should take into account the smoothness of the resulting surfaces. To enforce the surface smoothness constraint it is not sufficient to use, for example, the local disparity historgram (*e.g.*, the expectation of constant disparity would bias the resulting surface towards a frontal orientation). Rather, both the values of disparities as well as the locations of features giving rise to these values should be considered.

We have presented an integrated approach to extracting surfaces from stereo. Along with performing matching and interpolation, depth and ridge contours are detected so as to enforce surface smoothness everywhere except across such contours. The contours are constrained to be smooth. The approach thus integrates matching, contour detection and surface interpolation. These modules help exploit redunandcy of information present in the image [Brad82]. The integration approach is in contrast to existing stereo algorithms which complete the matching process before interpolating to obtain a dense depth map. As a consequence of integration, the computational effort is relatively uniformly distributed across the various integrated processes, unlike existing algorithms where most of the computation is devoted to feature point matching. The approach described is fairly domain independent since it uses no constraint other than the assumption of piecewise smoothness.

The integrated use of a surface representation has important advantages. First, the estimate of surface orientation can be used to predict the expected orientation shift of intensity edges. Second, the compression of the apparent size of a region can be predicted, so that the number of unmatchable points can be estimated. Third, by explicitly detecting contours, occluded regions can be identified, and mismatches in these regions can be avoided; further, it is possible to enforce smoothness of occluding and ridge contours. Fourth, it is possible to handle transparent surfaces. Finally, the availability of a surface represenation provides a way of combining the information in multiscale features. The algorithm works in a coarse-to-fine mode. It generates a progressively refined set of depth maps of a scene at increasing degrees of resolution. A given coarse level surface predicts the locations of edge matches at the next finer level. The matched features at the finer level provide a more refined surface which in turn predicts pairs of edges to be matched at the next finer level of resolution. Thus, the different "frequency channels" interact via the surface representation. There is no restriction placed on which channels can detect what ranges of disparities.

The approach described lends itself to a parallel implementation since the processing in different parts of the image can be carried out in parallel. In fact, the algorithm was run on a network of Sun workstations, by dividing up the image and letting each workstation process its own piece. There are also no major iterative steps. The results are usually accurate to within a pixel of disparity. The errors occur mainly around the occluding boundaries, apparently because of errors in the locations of the boundaries.

Problems

The contours found by the program are occasionally misplaced or missing, resulting in large disparity errors near occluding contours. The main reason for this is that the contours are detected and placed on the basis of local information, and the zero crossings in the vicinity of the contour may be sparse, or distorted by the blurring of

different regions across the contour. Since we expect contours in the real world to be smooth and continuous, the detection and location of contours should be done while enforcing this constraint. The present algorithm only partially enforces smoothness, by fitting cubic splines to the detected contour points.

The edges detected are inaccurate many times, causing errors in disparity. Usually this is not a problem because the surface patches are fit to many points, and the errors tend to cancel out. However, in some cases artifacts are created, *i.e.*, zero crossing contour segments which are present in one image but not in the other. This causes mismatched or unmatchable points. A smaller edge operator might give better localization, with fewer artifacts.

There are many places in the example images where the surface is "unknown," because patches could not be fit in the vicinity. The surface in those areas could be estimated from the coarser levels, if known. Alternatively, a more global interpolation could be done. Patches could be used from further away, or the surface could be interpolated using the depth points themselves over the entire surface.

Occasionally incorrect patches are fit, when the scene contains some periodic image structure. Locally the patches are a good fit to the data, but globally they are not. The current algorithm performs only local matching and interpolation, and so is unable to recognize the error. A solution to this is to use more global information in deciding which patch is correct.

Occasionally the coarse levels provide an incorrect disparity estimate to the fine levels, causing the fine level to be unable to match the points. This usually happens when the surface at the coarse level is extrapolated into an unknown area, so that an estimate can be provided for that area. Since there are no points to constrain the surface there, a small error in the parameters of the known surface patch can cause a large error in the extrapolated surface. One solution to this would be to use a wider matching window in an effort to match the points, although this would be computationally expensive.

Finally, human beings can fuse isolated line segments, which this algorithm would be unable to do. The simplest example would be a single vertical line segment in each image. The algorithm would not be able to fit a surface patch to the line segment, and so would not be able to match it. Isolated line segments are not uncommon in man-made environments, because man-made objects often have surfaces with little or no image texture, and the only intensity edges in the scene come from the contours of the object, or a rare surface marking.

REFERENCES

[Akim78] Akima, H., "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points," *ACM Trans. Math. Software*, vol. 4, no. 2, pp. 148-159, June 1978.

[Arno78] Arnold, R. D., "Local Context in Matching Edges for Stereo Vision," *Proc. of Workshop on Image Understanding*, pp. 65-72, Cambridge, May 1978.

[Ayac85] Ayache, N. and B. Faverjon, "Fast Stereo Matching of Edge Segments Using Prediction and Verification of Hypotheses," *Proc. of Computer Vision and Pattern Recognition* , June 1985, pp. 662-664.

[Bake81] Baker, H. H., *Depth from Edge and Intensity Based Stereo*, Ph.D. thesis, University of Illinois, Urbana, Illinois, 1981.

[Barn80] Barnard, S. T. and W. B. Thompson, "Disparity Analysis of Images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 4, pp. 333-340, July 1980.

[Barn82] Barnard, S. T. and M. A. Fischler, "Computational Stereo," *ACM Computing Surveys*, vol. 14, no. 4, pp. 553-572, December 1982.

[Berz84] Berzins, V., "Accuracy of Laplacian Edge Detectors," *Computer Vision, Graphics, and Image Processing*, vol. 27, pp. 195-210, 1984.

[Blin76] Blinn, J. F. and M. E. Newell, "Texture and Reflection in Computer Generated Images," *Communications of the ACM*, vol. 19, no. 9, October 1976.

[Brad82] Brady, M., "Artificial intelligence approaches to image understanding," in *Pattern Recognition Theory and Applications (Proc. of the NATO Advanced Study Institute, March 29 - April 10, 1981)*, J. Kittler, K.S. Fu, and L.F. Pau (Eds.), D. Reidel Publishing Co., Dordrecht, Holland, 1982.

[Brod56] Brodatz, P., *Textures: A Photograph Album for Artists and Designers*, Dover, New York, 1956.

[Burt80] Burt, P. and B. Julesz, "A Disparity Gradient Limit for Binocular Fusion," *Science*, vol. 208, pp. 615-617, May 1980.

[Duda73] Duda, R.O. and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[East85] Eastman, R. D., and A. M. Waxman, "Disparity Functional and Stereo Vision," *Proc. DARPA Image Understanding Workshop*, Miami Beach, pp. 245-254, December 9-10, 1985.

[Fole83] Foley, J. D. and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1983.

[Genn77] Gennery, D. B., "A Stereo Vision System for an Autonomous Vehicle," *Proc. 5th IJCAI*, pp. 576-582, August 1977.

[Gill84] Gilliam, B., T. Flagg, and D. Finlay, "Evidence for Disparity Change as the Primary Stimulus for Stereoscopic Processing," *Perception and Psychophysics*, vol. 36(6), pp. 559-564, 1984.

[Grim81] Grimson, W. E. L., *From Images to Surfaces*, MIT Press, Cambridge, 1981.

[Grim85] Grimson, W. E. L., "Computational Experiments with a Feature Based Stereo Algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 17-34, January 1985.

[Hann80] Hannah, M.J., "Bootstrap Stereo," *Proc. of Workshop on Image Understanding* , pp. 201-208, College Park, Md, April 1980.

[Hend79] Henderson, R.L., W.J. Miller, and C.B. Grosch, "Automatic Stereo Reconstruction of Man-made Targets," *Soc. P.I.E.*, vol. 186, no. 6, pp. 240-248, 1979.

[Hoff85] Hoff, W. A., and N. Ahuja, "Depth from Stereo," *Proc. Fourth Scandinavian Conference on Image Analysis*, June 18-20, Trondheim, Norway, pp. 761-768, 1985.

[Ito86] Ito, M. and A. Ishii, "Range and Shape Measurement Using Three-View Stereo Analysis," *Proc. of Computer Vision and Pattern Recognition*, June 1986, pp. 9-14.

[Jule71] Julesz, B., *Foundations of Cyclopean Perception*, Univ. of Chicago Press, 1971.

[Kim86] Kim, N. H. and A. C. Bovik, "A Solution to the Stereo Correspondence Problem Using Disparity Smoothness Constraint," *Proc. of IEEE Conf. Systems, Man, and Cybernetics*, Atlanta, October 1986.

[Lecl84] Leclerc, Y. and S. W. Zucker, "The Local Structure of Image Discontinuities in One Dimension," *Proc. 7th International Conf. Pattern Recognition*, pp. 576-582, August 1977.

[Luca82] Lucas, B. D., "Automatic Generation of Depth Maps from Stereo Images," *Proc. of Workshop on Image Understanding*, pp. 309-314, Palo Alto, September 1982.

[Marr79] Marr, D. and T. Poggio, "A Theory of Human Stereo Vision," *Proc. R. Soc. Lond.*, vol. B 204, pp. 301-328, 1979.

[Marr80] Marr, D. and E. Hildreth, "Theory of Edge Detection," *Proc. R. Soc. Lond.* , vol. B 207, pp. 187-217, 1980.

[Marr82] Marr, D., *Vision*, Freeman, San Francisco, 1982.

[Mayh80] Mayhew, J. E.W. and J. P. Frisby, "The Computation of Binocular Edges," *Perception*, vol. 9, pp. 69-86, 1980.

[Mayh81] Mayhew, J. E.W. and J. P. Frisby, "Psychophysical and Computational Studies towards a Theory of Human Stereopsis," *Artificial Intelligence*, vol. 17, pp. 349-385, August 1981.

[Medi85] Medioni, G. and R. Nevatia, "Segment-Based Stereo Matching," *Computer Vision, Graphics, and Image Processing*, vol. 31, pp. 2-18, July 1985.

[Mora81] Moravec, H. P., "Rover Visual Obstacle Avoidance," *Proc. 7th IJCAI*, vol. 2, pp. 785-790, Vancouver, August 1981.

[Nish81] Nishara, H.K. and N.G. Larson, "Towards a Real Time Implementation of the Marr and Poggio Stereo Matcher," *Proc. of Workshop on Image Understanding*, pp. 114-120, Washington D.C., April 1981.

[Ohta85] Ohta, Y. and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 139-154, March 1985.

[Pant78] Panton, D. J., "A Flexible Approach to Digital Stereo Mapping," *Photogramm. Eng. Remote Sensing*, vol. 44, no. 12, pp. 1499-1512, December 1978.

[Piet86] Pietikäinen, M. and D. Harwood, "Depth from Three Camera Stereo," *Proc. of Computer Vision and Pattern Recognition*, June 1986, pp. 2-8.

[Praz85] Prazdny, K., "Detection of Binocular Disparities," *Biological Cybernetics*, vol. 52, pp. 93-99, 1985.

[Rich77] Richards, W., "Stereopsis with and without Monocular Contours," *Vision Research*, vol. 17, pp. 967-969, 1977.

[Rose76] Rosenfeld, A., R. Hummel, and S. Zucker, "Scene Labeling by Relaxation Operations," *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-6, pp. 420-433.

[Terz83] Terzopoulos, D., "Multilevel Computational Processes for Visual Surface Reconstruction," *Computer Vision, Graphics, and Image Processing*, vol. 24, pp. 52-96, 1983.

[Yell76] Yellott, J. I., "Binocular Depth Inversion," *Scientific American*, pp. 152-159, April 1976.